MESHLESS METHOD FOR MODELING LARGE DEFORMATION WITH ELASTOPLASTICITY

by

JIANFENG MA

B.S., Beijing Institute of Light Industry, China, 1995
M.S., Beijing Institute of Technology, China, 1998

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2007

Abstract

Over the past two decades meshless methods have attracted much attention owing to their advantages in adaptivity, higher degree of solution field continuity, and capability to handle moving boundary and changing geometry. In this work, a meshless integral method based on the regularized boundary integral equation has been developed and applied to two-dimensional linear elasticity and elastoplasticity with small or large deformation.

The development of the meshless integral method and its application to two-dimensional linear elasticity is described first. The governing integral equation is obtained from the weak form of elasticity over a local sub-domain, and the moving least-squares approximation is employed for meshless function approximation. This formulation incorporates: a subtraction method for singularity removal in the boundary integral equation, a special numerical integration for the calculation of integrals with weak singularity which further improves accuracy, a collocation method for the imposition of essential boundary conditions, and a method for incorporation of natural boundary conditions in the system governing equation. Next, elastoplastic material behavior with small deformation is introduced into the meshless integral method. The constitutive law is rate-independent flow theory based on von Mises yielding criterion with isotropic hardening. The method is then extended to large deformation plasticity based on Green-Naghdi's theory using updated Lagrangian description. The Green-Lagrange strain is decomposed into the elastic and plastic part, and the elastoplastic constitutive law is employed that relates the Green-Lagrange strain to the second Piola-Kirchhoff stress. Finally, a pre- and post-processor for the meshless method using node- and pixel-based approach is presented. Numerical results from the meshless integral method agree well with available analytical solutions or finite element results, and the comparisons demonstrate that the meshless integral method is accurate and robust. This research lays the foundation for modeling and simulation of metal cutting processes.

MESHLESS METHOD FOR MODELING LARGE DEFORMATION WITH ELASTOPLASTICITY

by

JIANFENG MA

B.S., Beijing Institute of Light Industry, China, 1995
M.S., Beijing Institute of Technology, China, 1998

A DISSERTATION

submitted in partial fulfillment of the requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Mechanical Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2007

Approved by:

Co-Major Professor
Dr. Prakash Krishnaswami

Approved by:

Co-Major Professor
Dr. Jack Xin

Copyright

Abstract

Over the past two decades meshless methods have attracted much attention owing to their advantages in adaptivity, higher degree of solution field continuity, and capability to handle moving boundary and changing geometry. In this work, a meshless integral method based on the regularized boundary integral equation has been developed and applied to two-dimensional linear elasticity and elastoplasticity with small or large deformation.

The development of the meshless integral method and its application to two-dimensional linear elasticity is described first. The governing integral equation is obtained from the weak form of elasticity over a local sub-domain, and the moving least-squares approximation is employed for meshless function approximation. This formulation incorporates: a subtraction method for singularity removal in the boundary integral equation, a special numerical integration for the calculation of integrals with weak singularity which further improves accuracy, a collocation method for the imposition of essential boundary conditions, and a method for incorporation of natural boundary conditions in the system governing equation. Next, elastoplastic material behavior with small deformation is introduced into the meshless integral method. The constitutive law is rate-independent flow theory based on von Mises yielding criterion with isotropic hardening. The method is then extended to large deformation plasticity based on Green-Naghdi's theory using updated Lagrangian description. The Green-Lagrange strain is decomposed into the elastic and plastic part, and the elastoplastic constitutive law is employed that relates the Green-Lagrange strain to the second Piola-Kirchhoff stress. Finally, a pre- and post-processor for the meshless method using node- and pixel-based approach is presented. Numerical results from the meshless integral method agree well with available analytical solutions or finite element results, and the comparisons demonstrate that the meshless integral method is accurate and robust. This research lays the foundation for modeling and simulation of metal cutting processes.

Table of Contents

List of Figures

List of Tables

Acknowledgements

There are many people I would like to thank for achieving this goal. I would like to express my gratitude to my prestigious major professors Dr. Prakash Krishnaswami and Dr. X.J. Xin for giving me advice and guidance during these years I spent working on this research. Their unending drive, comprehensive knowledge on solid mechanics, mathematics and numerical computation and strong ability of analyzing and solving problem give me great impression. They offered invaluable mentorship not only for my research, but also for my future career. I would like to thank Professor Duy H. Hua of the Chemistry Department for serving as the chairperson of the examining committee and thank Dr. Shuting Lei and Dr. Daniel Swenson for serving on my committee and giving me valuable advice and suggestions.

I would like to thank Department of Mechanical and Nuclear Engineering at Kansas State University for giving me the opportunity to follow a graduate program here and apply for Doctorate degree.

I would like to thank my parents, my wife, my parents-in-law, my brothers, and my sisters for their constant encouragement and support. A special prayer goes to my mother, who did not live to see this day, but who is always with me.

I also would like to thank all good friends I got here in Manhattan. Especially I give thanks to Wenjie Liu and his family, Congjian Zhang, and Jianchu Yao.

Dedication

To my family

CHAPTER 1 - Introduction

Metal cutting is one of the most common manufacturing processes for producing parts of desired dimensions. It is used to remove unwanted material from a workpiece and obtain specified geometrical dimensions and surface finish. During metal cutting process, the work piece is subjected to large deformation with elastoplasticity at a high strain rate in the primary deformation area which extends from the tip of the cutting tool to the junction between surface of the undeformed work material and the deformed chip. In the meantime, the metal cutting process is characterized by continuous change in geometry and moving boundary as the cutting edge proceeds.

One of the state-of-the-art efforts in metal cutting is to use numerical models to simulate the cutting process to predict tolerance, cutting forces, chip formation, and residual stresses. These numerical models can greatly increase the understanding of the cutting process and reduce the number of experiments which are traditionally used for process selection, tool design, chip breakage investigations, and machinability evaluation.

The finite element method (FEM) is one of the most widely used computational methods in mechanics and has been applied to study and simulate metal cutting processes in the last two decades. The essential features of FEM are the discretization of the problem domain into elements and the use of local interpolation functions to represent a solution field that is globally or piecewise continuous. In FEM, the concept of an element is essential; it defines the connectivity between nodes. Because of the element, each node has a fixed number of nodes directly connected to it. This fixed nodal connectivity through elements, however, frequently makes it difficult to generate a good mesh and satisfy topological requirements simultaneously; in particular, robust and efficient three-dimensional (3D) mesh generators are yet to be developed. In the simulation of metal cutting, FEM method entails remeshing in each step as the cutting edge proceeds and frequently the large deformation in primary deformation region makes the mesh distort severely.

Over the past two decades, meshless methods have attracted much attention owing to their advantages such as improved adaptivity, higher degree of continuity in the solution field, and better handling of moving boundaries and changing geometry. In the meshless method, the

concept of an element is eliminated. The model geometry consists of a distribution of nodes over the domain and the approximate solution is constructed entirely in terms of the values of the field variables of interest at these nodes. Consequently, the interaction between nodes in the meshless method is much more flexible. Each node, through the use of localized weight functions, always influences (and is influenced by) the values of field variables at nodes that are nearby, but there is no limitation on the number of other nodes that can lie within the domain of influence of a particular node. This makes the model building in meshless methods much simpler than in FEM: adaptivity for meshless methods is easily achieved by adding or removing nodes without the troublesome remeshing of elements and enables meshless method to overcome the problems (moving boundary and severe distortion) encountered by using FEM in the simulation of metal cutting process.

In this project, we are going to develop meshless method to lay the foundation for modeling and simulation of metal cutting processes.

Several versions of meshless methods have been developed and these methods may be categorized in a number of ways. One possible categorization is by the type of integration domain. Based on this criterion, meshless methods can be classified as boundary methods or domain methods. The class of boundary methods includes the boundary node method (BNM) [1][2] and boundary point interpolation method (BPIM) [3], while the class of domain methods includes all other meshless methods.

Most methods, such as element-free Galekin (EFG), reproducing kernel particle method (RKPM), and boundary node method (BNM) are based on the weak form of the linear elasticity equations, defined over the whole problem domain. Such methods are “meshless” only in terms of the interpolation or approximation of the field or boundary variables; they still need to use a background mesh to integrate the weak form over the global domain or boundary. The requirement of a background mesh for integration makes these methods not truly meshless. Methods based on collocation over multiple local domains have the flexibility to choose simple local domains such as circles in 2D and spheres in 3D so that integration of the weak form is independent of the global domain. Thus, the need for a background mesh is eliminated and the integration and can be carried out easily. These methods may be called truly meshless because they do not require a background mesh either for the interpolation of the solution variables, or

for the integration of the energy. One example of a truly meshless method is the local boundary integral equation (LBIE) method [4][5][6][7].

The meshless method based on local boundary integral equation (LBIE) [4][5][6][7] introduces a companion solution and satisfies both the essential and the natural boundary conditions. In comparison with other methods, it has several advantages: (1) it is a truly meshless method; (2) the governing equations are derived from boundary integral equation which satisfies exactly the equilibrium equations, the constitutive law, and the kinematic relations of elasticity; (3) the requirements for the continuity of the trial function are greatly relaxed; and (4) no derivatives of shape functions are needed in constructing the system stiffness matrix for the internal nodes and the boundary nodes that are unrelated to essential boundary conditions. These aspects make the LBIE meshless method very attractive. There are, however, several serious deficiencies in the formulation presented in [5]. First, the governing integral equation contains a strong singularity (1/r type for a line integral); in this case, the direct limit approach as used in [5], usually leads to non-convergent results. Second, no special treatment is used for the weak singularity (logarithmic type), which makes it difficult to achieve high numerical accuracy. Third, the essential boundary conditions are not imposed explicitly or efficiently. These aspects should be improved before the LBIE meshless method can be regarded as an accurate and robust method.

Most development in meshless methods to date has been focused mainly on problems which are characterized by linear elastic material behavior with small deformations. Research in elastoplasticity with small deformation has been done by many researchers using FEM. Similarly, several researchers have studied elastoplastic problems with large deformation using FEM. However, the use of meshless methods to analyze elastoplastic behavior with either small deformation or large deformation has received very limited study, and is only now gaining attention. Studies [8][9][10][11][12] have used element-free Galerkin methods to solve elastoplastic problems with small deformation. Elastoplastic problems with large deformation have also been solved using element-free Galerkin methods [13][14][15][16]. As noted earlier, however, these are not truly meshless methods since they make use of a background mesh for integration.

In this work, a meshless integral method based on the regularized boundary integral equation is proposed. First, the method is developed and applied to two-dimensional linear

elasticity problems. This method is a truly meshless method and does not require a background mesh for integration. The governing integral equation is obtained from the weak form of elasticity over a local sub-domain, and the moving least-squares approximation is employed for meshless function approximation. It is built on the LBIE method proposed by Atluri et. al. [5], and incorporates improvements in three areas: (1) A subtraction method is used to remove the strong singularity in the local integral equation, which makes the proposed method practicable and accurate. (2) A special numerical integration scheme is employed for the calculation of integrals with weak singularity, which further improves accuracy. (3) The collocation method is employed to enforce essential boundary conditions, while the natural boundary conditions are incorporated in the system governing equation and require no special handling.

Next, elastoplastic material behavior with small deformation is introduced into the meshless integral method. The governing integral equation is obtained from the weak form of elastoplasticity with small deformation over a local sub-domain. The constitutive law that is used is from rate-independent flow theory based on the von Mises yield criterion with isotropic hardening. A fixed point iteration method is employed to solve the governing equations, which are nonlinear due to the elastoplastic nature of the material. This method has the following advantages: the derivative of the stiffness matrix is not needed, the formulation is simple, and the implementation is relatively easy. This meshless method can handle any prescribed loading profile, including unloading and reversed loading. Numerical examples show that the elastoplastic integral meshless method is accurate and robust.

The method is then extended to large deformation elastoplasticity using an updated Lagrangian description. Among the many theories that have been proposed for large deformation elastoplasticity, E.H. Lee's theory [17] and Green-Naghdi's theory [18] are the most relevant to our current development. Lee's theory begins with the exact kinematics of elastoplastic deformation and assumes a multiplicative decomposition of the deformation gradient into an elastic part and a plastic part. The limitations to this theory are: (1) the material under consideration has to be isotropic, and (2) the material obeys the isotropic hardening rule. Green-Naghdi's theory, on the other hand, starts with the decomposition of Green-Lagrange strain into an elastic part and a plastic part. This theory is more flexible because it can be applied to either anisotropic or isotropic material and its hardening rule can also be generalized. In addition, the computing procedure involved is relatively straightforward [19]. In this research, we use Green-

Naghdi's theory. For the constitutive law, the Green-Lagrange strain is decomposed into an elastic part and a plastic part. The Green-Lagrange strain is then related to the second Piola-Kirchhoff stress, and the yield function is constructed in the second Piola-Kirchhoff stress space. The incremental governing integral equation is obtained from the weak form of elastoplasticity with large deformation over a local sub-domain of the reference domain. For each load increment, the state variables are defined with respect to the reference configuration. At the end of each increment, the state variables are updated to the new converged configuration corresponding to this increment. This new configuration becomes the reference configuration for the next load increment.

To use a meshless method, the input file describing the problem of interest must be constructed; further the user may also need support for analyzing and visualizing the results obtained by the meshless method. Commercial software packages such as ANSYS [20], MSC/PATRAN [21] and CONPLOT [22] have these two functionalities, but these are element-based and cannot be used by meshless methods. Liu [23] developed a post-processor for a particular meshless method, but this method also makes use of a background mesh. In order to post-process the output file of the meshless method more conveniently and efficiently, we have developed a generic meshless pre-processor and post-processor for meshless solvers using node-based and pixel-based approaches, as opposed to an element-based approach.

The improved meshless integral method based on the regularized integral equation is accurate and robust, and appears remarkably promising. In the application of the meshless integral method to linear elasticity for which analytical solutions are available, for all the convergence patch tests, the meshless results agree with the exact solution to within machine accuracy, and for all other numerical tests (the cantilever beam problem and the infinite plate with a circle hole), the meshless results are in excellent or satisfactory agreement with analytical solution. In the application to elastoplastic problem with either small deformation or large deformation, because few closed-form solutions to realistic engineering problems are available, finite element results are used to validate the meshless results. For the elastoplastic problem with small deformation, excellent or satisfactory accuracy is shown in numerical tests (the constant patch tests, the sheat patch tests, the finite plate with a circular hole problem, and the thick-walled cylinder problem). For the elastoplastic problem with large deformation, numerical results show excellent accuracy.

In this research, no experiments are used to validate the numerical results. Instead, we use the analytical results published in the literature or the finite element results obtained using a commercial software, ANSYS, which is well established and trustworthy as the bases for comparison to validate the numerical results.

For linear elasticity (Chapter 2), the material behavior is fully characterized by Young's modulus and Poisson's ratio. Since Poisson's ratio is dimensionless and Young's modulus has the same unit as stresses do, through dimensionless analysis, the value of Young's modulus has no influence on the stresses normalized by Young's modulus. For elastoplasticity (Chapter3 and Chapter 4), we will use three types of metal for simulation: AISI 1020 steel, ASTM A514 structural steel, and AISI 1045 unalloyed carbon steel.

This thesis follows the alternate format accepted by Mechanical and Nuclear Engineering Department, which is organized as a collection of four papers, each of which is complete in itself. Chapter 2 is a paper describing the proposed meshless integral method and its application to two-dimensional linear elasticity. Chapter 3 details the introduction of small deformation elastoplasticity into the new meshless integral method. Chapter 4 introduces large deformation elastoplasticity into the new meshless integral method. The pre- and post- processors that were developed for meshless solvers are discussed in Chapter 5, and Chapter 6 presents the conclusions drawn from this work, along with suggestions for future developments in this field.

Reference

[1] Mukherjee, Y.X. and Mukherjee, S. Boundary node method for potential problems. International Journal for Numerical Methods in Engineering. 1997; 40: 797-815.

[2] Chati, M.K. Mukherjee, S and Mukherjee, YX. The boundary node method for three-dimensional linear elasticity. International Journal for Numerical Methods in Engineering. 1999; 46: 1163-1184.

[3] Gu, Y.T. and Liu, G.R. A boundary point interpolation method for stress analysis of solids. Computational Mechanics. 2002; 28: 47-54.

[4] Zhu T., Zhang J-D., and Atluri S.N. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Computational Mechanics. 1998; 21: 223-235.

[5] Atluri S.N., Sladeck J., Sladeck V., and Zhu T. The local boundary integral equation (LBIE) and its meshless implementation for linear elasticity. Computational Mechanics. 2000; 25: 180-198.

[6] Sladek, J., Sladeck, V., and Van Keer, R. Meshless local boundary integral equation for 2D elastodynamic problems. International Journal for Numerical Methods in Engineering. 2003: 235-249.

[7] Long, S. and Zhang. Q. Analysis of thin plates by the local boundary integral equation (LBIE) method. Engineering Analysis with Boundary Elements. 2002; 26: 707-718.

[8] Xu Y, Saigal S, An element-free Galerkin analysis of steady dynamic growth of a mode I crack in elastic-plastic materials, International Journal of Solids and Structures, 1999, 36, 1045-1079.

[9] Rao, B.N. and Rahman, S. An enriched meshless method for non-linear fracture mechanics. International Journal for Numerical methods in Engineerig. 2004; 59: 197-223.

[10] Kargarnovin, M.H., Toussi, H.E., and Fariborz, S.J. Elasto-plastic element-free Galerkin method. Computational Mechanics. 2004; 33: 206-214.

[11] Xu, Y. and Saigal, S. Element free Galerkin study of steady quasi-static crack growth in plane strain tension in elastic-plastic materials. Computational Mechanics. 1998; 22: 206-214.

[12] Chen, Y.P., Lee, J.D., and Eskandarian, A. Dynamic meshless method applied to nonlocal crack problems. Theoretical and Applied Mechanics. 2002: 38: 293-300.

[13] Belytschko, T., Krysl, P., and Krongauz, Y. A three-dimensional explicit element-free Galerkin method. International Journal for Numerical methods in Fluids. 1997; 24: 1253-1270.

[14] Rossi, R. and Alves, M. K. On the analysis of an EFG method under large deformations and volumetric locking. Computational Mechanics. 2007; 39: 381-399

[15] Chen, Y. P., Eskandarian, A., Oskard, M., and Lee, J. D. Meshless analysis of high-speed impact. Theoretical and Applied Fracture Mechanics. 2005; 44: 201-207.

[16] Eskandarian, A., Chen, Y. P., Oskard, M., and Lee, J. D. Meshless analysis of fracture. Plasticity and impact. Proceedings of IMECE'03. 2003 ASME International Mechanical Engineering Congress. Washington, D.C., November 15-21, 2003.

[17] Lubarda, V. A. and E. H. Lee. A correct definition of elastic and plastic deformation and its computational significance. J. appl. Mech. 1981; 48: 35-40.

[18] Green, A. E and Naghdi, P.M. A general theory of an elastic-plastic continuum. Archs Rational Mech. Anal. 1965; 18: 251-281.

[19] Chiou, J. H., Lee, J. D., and Erdman, A. G. Comparison between two theories of plasticity. Computers & Structures. 1986: 24(1): 23-37.

[20] ANSYS 5.6 Documentation, ANSYS Incorporated, 1999.

[21] The MacNeal-Schwendler Corporation, MSC/PATRAN user's manual. Los Angeles, CA: The MacNeal-Schwendler Corporation, 1996.

[22] James, Mark. A Finite Element Model of Fluid Flow in Jointed Rock. Master thesis. Kansas State University. 1988.

[23] MFree2D Software. http://www.nus.edu.sg/ACES/software/meshless2D.htm. 1999.

CHAPTER 2 - A Meshless Integral Method Based on Regularized Boundary Integral Equation

Published in:

Computer Methods in Applied Mechanics and Engineering 2006; 195: 6258-6286

Authors' Names:

Anthony Bodin, Jianfeng Ma, X.J. Xin, and Prakash Krishnaswami

Authors' Affiliations

Department of Mechanical and Nuclear Engineering, Kansas State University, 302 Rathbone Hall, Manhattan, KS 66506-5205

2.1 Abstract

A meshless integral method based on the regularized boundary integral equation is developed and applied to two-dimensional linear elasticity. The governing integral equation is obtained from the weak form of elasticity over a local sub-domain, and the moving least-squares approximation is used for meshless function approximation. The method is built on the LBIE method proposed by Atluri and coworkers, and several key improvements are introduced in this work that significantly enhance the accuracy and robustness of the method. The most critical improvement is the use of the subtraction technique to remove the strong singularity that results in a regularized governing integral equation. The technique is straightforward and efficient, and is much simpler and easier compared to other singularity removal techniques. A special numerical integration is employed for the calculation of integrals with weak singularity which further improves accuracy. The collocation method is employed to enforce the essential boundary conditions exactly, which is simple and computationally efficient. The natural boundary conditions are incorporated in the system governing equation and require no special handling. Numerical tests show that the meshless integral method is accurate and robust. The

effects of weight function, support domain, sub-domain, and monomial basis are investigated and discussed.

KEY WORDS: meshless method, local boundary integral equation, moving least-squares approximation, subtraction method, singularity removal, linear elasticity.

2.2 Introduction

The finite element method (FEM) (e.g., [1][2]) has been one of the most widely used computational methods in mechanics. The essential features of FEM are the discretization of the problem domain into elements and the use of local interpolation functions to represent a solution field that is globally or piecewise continuous. In FEM, the concept of an element is essential; it defines the connectivity between nodes. Because of the element, each node has a fixed number of nodes directly connected to it. This fixed nodal connectivity through elements, however, frequently makes it difficult to generate a good mesh and satisfy topological requirements simultaneously; robust and efficient three-dimensional (3D) mesh generators are yet to be developed. Over the past two decades the meshless methods have attracted much attention owing to their advantages in adaptivity, higher degree of continuity in the solution field, and capability to handle moving boundary and changing geometry. In the meshless method, the concept of an element is eliminated. The model geometry consists of a distribution of nodes over the domain and the approximate solution is constructed entirely based on these nodes. Consequently, the nodal connectivity in the meshless method is much more flexible. Each node, through the use of a localized weight function, is always connected to the nodes that are nearby, and there is no limitation on the number of other nodes a node can be connected to directly. This makes the model building in meshless methods much simpler than in FEM: adaptivity for meshless methods is easily achieved by adding or removing nodes without the troublesome remeshing of elements.

Several versions of meshless methods have been developed. The initial idea of meshless methods dates back to the Smoothed Particle Hydrodynamics method (SPH) [3] for modeling astrophysical phenomena in the 1970's. Later developments include Diffuse Element Method (DEM) [4], Element-Free Galerkin method (EFG) [5][6][7][8], Reproducing Kernel Particle Method (RKPM) [9][10][11][12], HP-Meshless Cloud method [13][14], Local Boundary Integral Equation (LBIE) method [15][16][17][18], meshless local Petrov-Galerkin (MLPG) method

[19][20][21][22], boundary node method (BNM) [23][24], natural element method [25], boundary point interpolation method (BPIM) [26], local point interpolation method (LPIM) [27], point interpolation method [28], point assembly method (PAM) [29], and so on. In DEM [4], the usual FEM interpolation is replaced by a diffuse approximation through the use of a least-squares approximation. The method has been applied to two-dimensional (2D) problems in potential theory and linear elasticity. The main idea of EFG method is to use moving least-squares approximation (MLSA) to construct the trial functions used in the Galerkin weak form. The EFG method was applied to 2D problems in linear elasticity and heat conduction [5], fracture mechanics [30], crack propagation [31], thin plate [7], wave propagation and dynamic fracture [8], elasto-plastic fracture mechanics [32], and shape sensitivity analysis and optimization [33]. The BNM involves a coupling between MLSA and boundary integral equations, and has been used for solving problems in both potential theory and elasticity [23][24].

The meshless methods may be categorized in a number of ways. One possible categorization is by the type of integration domain: (1) boundary type methods such as the boundary node method (BNM) [23][24] and boundary point interpolation method (BPIM) [26]; and (2) domain type methods which include all other meshless methods. Most methods such as EFG, RKPM, and BNM are based on the weak form defined over the whole problem domain. Such methods are "meshless" only in terms of the interpolation or approximation of the field or boundary variables; they still need to use a background mesh to integrate the weak form over the global domain or boundary. The requirement of a background mesh for integration makes these methods not truly meshless. Methods based on collocation over multiple local domains have the flexibility to choose simple local domains such as circles in 2D and spheres in 3D so that integration of the weak form is independent of the global domain (and hence no need for a background mesh) and can be carried out easily. These methods may be called truly meshless because they do not require a background mesh either for the purpose of interpolation of the solution variables, or for the integration of the energy. Examples of truly meshless methods include MLPG method [19][20][21][22], LBIE method [15][16][17][18], and local point interpolation method (LPIM) [27].

The meshless method based on local boundary integral equation (LBIE) [15][16][17][18] utilizes the boundary integral representation over a local contour and the moving least-squares approximation for function approximation. With the introduction of a companion solution, the

method satisfies the essential as well as the natural boundary conditions. In comparison with other methods, it has several advantages: (1) it is a truly meshless method and does not need a background mesh; (2) the governing equations are derived from boundary integral equation which satisfies exactly the equilibrium, the constitutive law, and the kinematic relation of elasticity; (3) the requirements for the continuity of the trial function are greatly relaxed; and (4) no derivatives of shape functions are needed in constructing the system stiffness matrix for the internal nodes and boundary nodes unrelated to essential boundary conditions. These aspects make the LBIE meshless method very attractive. There are, however, several serious deficiencies in the formulation presented in [16]. First, the governing integral equation contains a strong singularity (1/r type for line integral) which converges only in the Cauchy Principal Value (CPV) sense. Direct limit approach with a finite cut of the singular core, as has been used in [16], usually leads to non-convergent results, i.e., the results depend on the size of the cut. Second, no special treatment is used for the weak singularity (logarithmic type), which makes it difficult to achieve high numerical accuracy. Third, the essential boundary conditions are not imposed explicitly or efficiently. The results for the numerical tests reported in [16] appeared to be sensitive to the values of several geometric parameters used in MLSA, such as the size of support function and, more notably, the scaling factor ($s^{<a>}$) in the Gaussian weight function, while in principle for a robust method the solutions should be insensitive to these parameters within reasonable ranges. There were test cases where the errors ran from below 0.1% to almost 1000%. Furthermore, for some cases, while the geometric parameters were varied over a certain range, the numerical results showed no consistent trend of convergence. These aspects should be improved before the LBIE meshless method can be regarded as an accurate and robust method.

The deficiencies described above are primarily caused by the inadequate numerical treatment used in [16] for handling singularities in the boundary integrals. In this work, we have developed numerical techniques that effectively eliminate these deficiencies. The subtraction method is used to derive the regularized governing integral equations in which the strong singularity (1/r type for line integrals) is removed. The approach devised is much more accurate than the direct limit approach used in [16]. The procedure is straightforward and also simpler and easier to implement than the lengthy singularity removal treatment proposed in [34]. A special numerical integration is employed for the calculation of integrals with weak singularity (logarithmic type). A collocation method is employed for enforcing the essential boundary

conditions exactly and easily, which requires almost no additional computational cost. The improved meshless integral method based on the regularized integral equation is accurate and robust, and appears remarkably promising. For the convergence patch tests, the numerical results agree with the exact solution to within machine accuracy. For all other tests, the numerical results are in excellent or satisfactory agreement with analytic solutions.

The structure of this paper is as follows: in Section 2.3, the local boundary integral equation for elasticity is derived for a sub-domain embedded in the original problem domain. Although the shape of the sub-domain can be arbitrary, a sphere (a disk in 2D) is used for simplicity in the meshless implementation. In Section 2.4, methods for evaluating strongly singular integrals are reviewed and discussed. The subtraction method is used in Section 2.5 to remove the strong singularity that is present in the local boundary integral equation. Section 2.6 summarizes the moving least-squares approximation (MLSA) for approximating the solution variables continuously over the problem domain. The meshless implementation of the regularized boundary integral equation using MLSA is presented in Section 2.7. Section 2.8 discusses the treatment of weak singularity, and Section 2.9 is devoted to the review and discussion of enforcement of essential boundary conditions which is one of the major difficulties of the meshless methods. The enforcement of natural boundary conditions is also described. Numerical examples are presented in Section 2.10 to assess the accuracy and effectiveness of the method. Discussion and conclusions from this study are given in Section 2.11.

2.3 Local Boundary Integral Equation for Linear Elasticity

Consider an elastic body represented by a domain Ω with boundary Γ. The governing elasticity equations are as follows:

$$\forall \mathbf{x} \in \Omega, \begin{cases} \sigma_{ij,j}(\mathbf{x}) + b_i(\mathbf{x}) = 0 \\ \varepsilon_{ij}(\mathbf{x}) = \frac{1}{2}\left(u_{i,j}(\mathbf{x}) + u_{j,i}(\mathbf{x})\right) \\ \sigma_{ij}(\mathbf{x}) = C_{ijkl}\varepsilon_{kl}(\mathbf{x}) \end{cases} \tag{1}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ (bold face denotes vectors or tensors) are the stress and the strain tensor associated with the displacement $\mathbf{u}$, $\mathbf{b}$ is the body force in the domain Ω, and $\mathbf{C}$ is the elastic stiffness matrix. An index i following a comma designates partial differentiation with respect to

x_i, and repeated indices indicate summation over the dimensionality of the problem. The essential and the natural boundary conditions on the boundary Γ are respectively:

$$u_i = \bar{u}_i \text{ on } \Gamma_u \tag{2}$$

$$t_i = \sigma_{ij} n_j = \bar{t}_i \text{ on } \Gamma_t \tag{3}$$

here, $\bar{\mathbf{u}}$ represents the prescribed displacement on Γ_u, $\bar{\mathbf{t}}$ represents the prescribed traction on Γ_t; $\mathbf{n}$ is the outward unit normal to the boundary; $\Gamma_t \cup \Gamma_u = \Gamma$ and $\Gamma_t \cap \Gamma_u = \varnothing$.

The weak form of (1) over a local domain is:

$$\int_{\Omega^{\langle a \rangle}} \left(\sigma_{ij,j}(\mathbf{x}) + b_i(\mathbf{x})\right) g_i\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) d\Omega(\mathbf{x}) = 0 \tag{4}$$

where the notation $< \cdot >$ in this paper is used to denote a node (e.g., <a> indicates node a, <b> indicates node b, etc.) in order to reserve the usual subscripts, i, j, etc., for denoting degree of freedom (DOF) components, $\Omega^{<a>} \subset \Omega$ is a sub-domain related to node <a>, $\mathbf{y}^{\langle a \rangle}$ is the position vector of node a which is also called a source point, $\mathbf{x}$ is the integration or field point which may or may not coincide with a node, and g_i is the test function. In the following, the functional dependence on $\mathbf{x}$, i.e. "$(\mathbf{x})$", will be dropped for brevity when no ambiguity is caused. If the fundamental solution of a point force in an infinite body is taken as the test function, the conventional boundary integral equation will result. The drawback of this approach is that the derivatives of the displacement field, which are cumbersome to compute numerically, are needed in the integral containing the traction term t_j on the boundary integral. To avoid such numerical difficulties, following Atluri et al. [15][16], we use a special test function defined by

$$g_i = \tilde{u}^*_{ji} e_j = \left(u^*_{ji} - \tilde{u}_{ji}\right) e_j \tag{5}$$

where e_j represents the j-th component of a unit force vector, u^*_{ji} is the fundamental solution of elasticity (i.e., the displacement field of an infinite body subjected to a unit force), $\tilde{u}_{ji}$ is the companion solution, to be given below, while $\tilde{u}^*_{ji}$ is the special test function.

The stress field σ^*_{ji} related to the fundamental solution u^*_{ji} satisfies the following equation:

$$\sigma^{*}_{ij,j}(\mathbf{x},\mathbf{y}^{<a>}) + \delta(\mathbf{x},\mathbf{y}^{<a>})\, e_i = 0 \tag{6}$$

where $\delta(\mathbf{x},\mathbf{y}^{<a>})$ is the Dirac delta function. The companion stress field $\tilde{\sigma}_{ji}$ related to the companion solution $\tilde{u}_{ji}$ satisfies the following equation:

$$\begin{cases} \tilde{\sigma}_{ij,j} = 0 & \text{in } \left|\mathbf{x} - \mathbf{y}^{<a>}\right| < h_s^{<a>} \\ \tilde{u}_{ji} = u^{*}_{ji} & \text{on } \left|\mathbf{x} - \mathbf{y}^{<a>}\right| = h_s^{<a>} \end{cases} \tag{7}$$

where $h_s^{<a>}$ is a constant. By construction, the special test function $\tilde{u}^{*}_{ji}$ is zero on the circle of radius $h_s^{<a>}$ centered at $\mathbf{y}^{<a>}$. The displacement fundamental solution u^{*}_{ij} and the associated traction t^{*}_{ij} have been presented (for example, [35]):

$$u^{*}_{ij}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) = \frac{1}{8\pi\mu(1-\bar{\nu})}\left\{\left[\left(4\bar{\nu}-3\right)\ln r^{\langle a\rangle}\right]\delta_{ij} + \frac{r_i^{\langle a\rangle} r_j^{\langle a\rangle}}{r^{\langle a\rangle 2}}\right\} \tag{8}$$

$$\begin{aligned} t^{*}_{ij}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) = -\frac{1}{4\pi(1-\bar{\nu})}\Bigg\{&\left[\frac{1}{r^{\langle a\rangle}}\frac{\partial r^{\langle a\rangle}}{\partial n^{\langle a\rangle}}\left(\left(1-2\bar{\nu}\right)\delta_{ij} + 2\frac{r_i^{\langle a\rangle} r_j^{\langle a\rangle}}{r^{\langle a\rangle 2}}\right)\right] \\ &- \frac{1-2\bar{\nu}}{r^{\langle a\rangle 2}}\left(r_i^{\langle a\rangle} n_j^{\langle a\rangle} - r_j^{\langle a\rangle} n_i^{\langle a\rangle}\right)\Bigg\} \end{aligned} \tag{9}$$

where $r^{<a>}$ is the distance from $\mathbf{y}^{\langle a\rangle}$ to $\mathbf{x}$; $r_i^{<a>} = x_i - y_i^{<a>}$; $\mathbf{n}^{<a>}$ is the outward unit normal to the boundary $\partial\Omega^{\langle a\rangle}$ at $\mathbf{x}$; $\bar{\nu} = \nu$ and $\bar{E} = E$ for plane strain, or $\bar{\nu} = \nu/(1+\nu)$ and $\bar{E} = E\left(1-\bar{\nu}^2\right)$ for plane stress; E is the Young's modulus, ν is the Poisson's ratio, and $\mu = \bar{E}/\left(2(1+\bar{\nu})\right)$ is the shear modulus.

The companion solutions are given by [16]:

$$\tilde{u}_{ij}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) = \frac{1}{8\pi\mu(1-\bar{\nu})}\left\{\left[\left(4\bar{\nu}-3\right)\ln h_s^{<a>} + \frac{5-4\bar{\nu}}{2\left(3-4\bar{\nu}\right)}\left(1 - \frac{r^{\langle a\rangle 2}}{h_s^{<a>2}}\right)\right]\delta_{ij} + \frac{r_i^{\langle a\rangle} r_j^{\langle a\rangle}}{h_s^{<a>2}}\right\} \tag{10}$$

$$\tilde{t}_{ij}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) = \frac{1}{4\pi(1-\bar{\nu})}\left\{\frac{3r_i^{\langle a\rangle} n_j^{\langle a\rangle} - r_j^{\langle a\rangle} n_i^{\langle a\rangle} - r_k^{\langle a\rangle} n_k^{\langle a\rangle}\delta_{ij}}{(3-4\bar{\nu})h_s^{<a>2}}\right\} \tag{11}$$

The special test functions are then obtained:

$$\tilde{u}^*_{ij}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)=\frac{1}{8\pi\mu(1-\bar{\nu})}\left\{\left[\left(4\bar{\nu}-3\right)\ln\frac{r^{\langle a\rangle}}{h_s^{<a>}}-\frac{5-4\bar{\nu}}{2\left(3-4\bar{\nu}\right)}\left(1-\frac{r^{\langle a\rangle 2}}{h_s^{<a>2}}\right)\right]\delta_{ij}+\left(1-\frac{r^{\langle a\rangle 2}}{h_s^{<a>2}}\right)\frac{r_i^{\langle a\rangle}r_j^{\langle a\rangle}}{r^{\langle a\rangle 2}}\right\} \tag{12}$$

$$\tilde{t}^*_{ij}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)=-\frac{1}{4\pi(1-\bar{\nu})}\left\{\left[\frac{1}{r^{\langle a\rangle}}\frac{\partial r^{\langle a\rangle}}{\partial n^{\langle a\rangle}}\left(1-2\bar{\nu}\right)-\frac{r_k^{\langle a\rangle}n_k^{\langle a\rangle}}{(3-4\bar{\nu})h_s^{<a>2}}\right]\delta_{ij}+2\frac{\partial r^{\langle a\rangle}}{\partial n^{\langle a\rangle}}\frac{r_i^{\langle a\rangle}r_j^{\langle a\rangle}}{r^{\langle a\rangle 3}}-\frac{1-2\bar{\nu}}{r^{\langle a\rangle 2}}\left(r_i^{\langle a\rangle}n_j^{\langle a\rangle}-r_j^{\langle a\rangle}n_i^{\langle a\rangle}\right)+\frac{3r_i^{\langle a\rangle}n_j^{\langle a\rangle}-r_j^{\langle a\rangle}n_i^{\langle a\rangle}}{(3-4\bar{\nu})h_s^{<a>2}}\right\} \tag{13}$$

With $\tilde{u}^*_{ji}$ as the test function, application of integration by parts to (4) twice while taking into account (6) and (7) leads to:

$$\begin{aligned}&-\int_{\Omega^{<a>}}\delta\left(\mathbf{x},\mathbf{y}^{<a>}\right)\;u_i(\mathbf{x})\;d\Omega(\mathbf{x})+\int_{\partial\Omega^{<a>}}\tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})\;t_j(\mathbf{x})\;d\Gamma(\mathbf{x})\\&-\int_{\partial\Omega^{<a>}}\tilde{t}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})\;u_j(\mathbf{x})\;d\Gamma(\mathbf{x})+\int_{\Omega^{<a>}}b_j(\mathbf{x})\;\tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})\;d\Omega(\mathbf{x})=0\end{aligned} \tag{14}$$

where $\tilde{t}^*_{ij}$ is the traction field associated with $\tilde{u}^*_{ji}$, and $\partial\Omega^{\langle a\rangle}$ is the boundary of $\Omega^{\langle a\rangle}$. In Equation (14) the integrals converge in the sense of the Cauchy Principal Value (CPV) as the traction field $\tilde{t}^*_{ij}$ exhibits 1/r type singularity.

Since the special test function $\tilde{u}^*_{ji}$ is zero on $\left|\mathbf{x}\text{-}\mathbf{y}^{<a>}\right|=h_s^{<a>}$, if $\Omega^{<a>}$ is taken as a disk in 2D or sphere in 3D with $\left|\mathbf{x}\text{-}\mathbf{y}^{<a>}\right|=h_s^{<a>}$ as its boundary, the boundary integral over $\partial\Omega^{\langle a\rangle}$ in equation (14) that contains the product of $\tilde{u}^*_{ji}$ and t_i (expressed in terms of the derivatives of u_i) will vanish. This significantly simplifies the numerical treatment and improves the accuracy and efficiency of the method.

An equation equivalent to (14) was first derived by Atluri et al [16]. Equation (14) in its current form cannot be used directly in numerical calculation because when $\mathbf{y}^{\langle a\rangle}$ is a boundary node, the equation contains strong singularity (1/r type in line integral) in the traction term $\tilde{t}^*_{ij}$. Almost all boundary-type meshless methods [23][24][26][15][16][17][18] encounter this type of

problem. Direct limit approach with a finite cut of the singular core, as has been used in [16], usually leads to non-convergent results, i.e., the results depend on the size of the cut. For the local integral equation approach to be a valid numerical method, the strong singularity must be handled appropriately.

2.4 Methods for Evaluating Strongly Singular Integrals

For the displacement boundary integral equations, there are usually two types of singularities in the kernels (in our case, $\tilde{u}^*_{ji}$ and $\tilde{t}^*_{ij}$) when the field point approaches the source point: the weak singularity which behaves ln(r) in line integral (2D elasticity) or 1/r in surface integral (3D elasticity), and the strong singularity which behaves 1/r in line integral or $1/r^2$ in surface integral. The weak singularity is actually removable, and can be evaluated satisfactorily by special Gaussian integration methods with adjusted weights [36][37][38], or mapping methods [39][40][41][42]. The special Gaussian integration method is derived by factoring the integrand into a non-singular base function and a singular weighting function such as 1/r or ln(r). When applicable, the special Gaussian integration method is accurate, efficient, and easy to use. However, this approach has two limitations: (1) the integrand must contain a factor in the exact singular weighting function form for which the special integration formula is derived; (2) the domain of integration usually must be of a certain shape such as a rectangle. The basic idea for mapping methods is to map the physical domain into a different domain in such a way that the integrand, after absorbing the Jacobian, becomes regular. Examples of mapping methods include the polar coordinate mapping [40], mapping of a curved quadratic triangle into a square [41], and mapping of triangle into a rectangle [42].

The strong singularity is more difficult to handle, as the singularity is real (i.e., cannot be removed by variable transformation or domain mapping) and not integrable (an example of integrable singularity is $\int_0^1 r^{-\beta} dr$ where $0 < \beta < 1$), and converges only in the sense of CPV. Several techniques have been developed for the numerical evaluation of strongly singular integrals: (1) subtraction method [43][44][45], (2) use of special solutions [46], (3) direct evaluation of CPV [47][48]. The basic idea of subtraction may be traced back to the time when Cauchy investigated the convergence of the complex Cauchy integrals by expressing $\int \frac{f(z)}{z - z_0} dz$

as $\int \frac{f(z)-f(z_0)}{z-z_0}dz + f(z_0)\int \frac{dz}{z-z_0}$, and this technique for regularizing singularities has been applied to problems in potential theory, elasticity, and acoustics. In essence, the subtraction methods involve first subtracting a term from the singular integral to make it regular and easily integrable by numerical methods, and then adding this term back which can usually be integrated analytically [43][44][45]. The observation behind the use of special solutions to evaluate singular integrals [46] is that, when the boundary integral equation is applied to some special, usually simplified situations such as rigid body translation in elasticity, all regular or weakly singular integrals can be evaluated accurately either in closed form or numerically, and the strongly singular integral can then be expressed in terms of the readily calculated results. This method in fact avoids the direct computation of singular integrals. In the direct evaluation of CPV [47][48], the CPV integrals defined on curved contours or surfaces are converted to regular integrals plus some simplified singular integrals. The regular integrals are computed using the usual Gaussian quadrature formulas, and the singular terms are integrated analytically.

A direct limit approach with a finite cut of the singular core was used in [16] for computing CPV integrals. As the singular term is not evaluated analytically, the accuracy is unsatisfactory, and the results appear to be non-convergent, i.e., the results depend on the size of the cut. Such a brute force evaluation is inappropriate for evaluating CPV. In [34], the direct limit approach for CPV integrals is combined with an optimal transformation of the integration variable to recast the integrands into smooth functions. The treatment leads to an improved accuracy over the direct limit approach, but the formulation is lengthy and cumbersome.

2.5 Regularized Local Boundary Integral Equation Using Subtraction Method

The subtraction technique is employed in the present study to remove the strong singularity, for which a limiting process is performed as discussed below. For simplicity in implementation, the local sub-domain is always chosen as a sphere or part of a sphere centered on a node. If node $\mathbf{y}^{\langle a\rangle}$ is an interior node, we can always adjust the radius of the sphere, $h_S^{\langle a\rangle}$, so that $\Omega^{\langle a\rangle}$ will stay fully inside Ω. If $\mathbf{y}^{\langle a\rangle}$ is a boundary node, it becomes impossible for a spherical $\Omega^{\langle a\rangle}$ to remain fully inside Ω for any size $h_S^{\langle a\rangle}$. In this case, the sub-domain $\Omega^{\langle a\rangle}$ will be the intersection of Ω and a sphere $\Omega'^{\langle a\rangle}$ of radius $h_S^{\langle a\rangle}$, centered at the boundary node, while

the boundary $\partial\Omega^{\langle a\rangle}$ is the union of the part of $\partial\Omega'^{\langle a\rangle}$ inside Ω and the part of $\partial\Omega$ inside $\Omega'^{\langle a\rangle}$, as illustrated in Figure 2.1. A further modification, which is necessary for handling the strong singularity, is the exclusion of a tiny sphere Ω_Δ of radius Δ (which later tends to zero) centered on $\mathbf{y}^{\langle a\rangle}$. Figure 2.2 shows schematically this modification.

We now decompose the boundary $\partial\Omega^{\langle a\rangle}$ into the following sections:

$$\partial\Omega^{\langle a\rangle} = C^{\langle a\rangle} \cup C_\Delta{}^{\langle a\rangle} \cup \Gamma^{\langle a\rangle} \tag{15}$$

$$\Gamma^{\langle a\rangle} = \Gamma_u{}^{\langle a\rangle} \cup \Gamma_t{}^{\langle a\rangle} \tag{16}$$

where $C^{<a>}$ is the circular part of $\partial\Omega^{<a>}$ of radius $h_S^{\langle a\rangle}$, C_Δ is the circular part of $\partial\Omega_\Delta$ of radius Δ, $\Gamma_u{}^{\langle a\rangle}$ is the section of $\Gamma^{\langle a\rangle}$ where the displacement is prescribed and $\Gamma_t{}^{\langle a\rangle}$ the section where the traction is prescribed. For interior nodes, $\Gamma^{\langle a\rangle}$ is zero, and $C^{<a>}$ is a full circle.

From construction, $\mathbf{y}^{\langle a\rangle}$ is outside $\Omega^{\langle a\rangle}$. Hence, the first integral containing $\delta\left(\mathbf{x},\mathbf{y}^{<a>}\right)$ in (14) is zero because of the properties of Dirac delta function. Applying the decomposition of equation (15) to (14), we obtain:

$$\begin{aligned}
&\int_{C^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) t_j(\mathbf{x})\,d\Gamma + \int_{C_\Delta{}^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) t_j(\mathbf{x})\,d\Gamma + \int_{\Gamma^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) t_j(\mathbf{x})\,d\Gamma \\
&- \int_{C^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) u_j(\mathbf{x})\,d\Gamma - \int_{C_\Delta{}^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) u_j(\mathbf{x})\,d\Gamma - \int_{\Gamma^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) u_j(\mathbf{x})\,d\Gamma \\
&+ \int_{\Omega^{\langle a\rangle}} b_j(\mathbf{x})\,\tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) d\Omega = 0
\end{aligned} \tag{17}$$

As $\mathbf{y}^{\langle a\rangle}$ lies outside $\Omega^{\langle a\rangle}$, we have

$$\int_{\partial\Omega^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) d\Gamma = 0 \tag{18}$$

which is equivalent to

$$\int_{C_\Delta{}^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) d\Gamma = -\int_{C^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) d\Gamma - \int_{\Gamma^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) d\Gamma \tag{19}$$

Equation (19) can also be obtained by applying (17) to rigid body motion in which $u_j = \text{const.}$, $t_j = 0$ and $b_j = 0$. Making use of the fact that the special test function $\tilde{u}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right)$ vanishes over $C^{\langle a\rangle}$, that $\int_{C_\Delta^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) t_j(\mathbf{x}) d\Gamma \to 0$ when $\Delta \to 0$, and that $u_i(\mathbf{x}) \to u_i\left(\mathbf{y}^{\langle a\rangle}\right) = u_i^{\langle a\rangle}$ on $C_\Delta{}^{\langle a\rangle}$ when $\Delta \to 0$, and subtracting (19) from (17), we obtain, in the limit of $\Delta \to 0$, the following:

$$\begin{aligned} &\int_{\Gamma^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) t_j(\mathbf{x}) d\Gamma + \int_{\Omega^{\langle a\rangle}} b_j(\mathbf{x}) \tilde{u}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) d\Omega \\ &- \int_{C^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right)\left(u_j(\mathbf{x}) - u_j^{\langle a\rangle}\right) d\Gamma - \int_{\Gamma^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right)\left(u_j(\mathbf{x}) - u_j^{\langle a\rangle}\right) d\Gamma = 0 \end{aligned} \tag{20}$$

On $C^{\langle a\rangle}$ where $r^{\langle a\rangle} = h_S^{\langle a\rangle}$, $\tilde{t}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right)$ given by (13) simplifies to:

$$\tilde{t}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) = -\frac{1}{2\pi(3-4\bar{\nu}) h_S^{\langle a\rangle}} \left\{(1-4\bar{\nu})\delta_{ij} + 4 n_i^{\langle a\rangle} n_j^{\langle a\rangle}\right\} \tag{21}$$

The integration of $\tilde{t}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right)$ over $C^{\langle a\rangle}$ can be obtained in closed form:

$$\int_{C^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) d\Gamma = -\alpha_{ij}^{\langle a\rangle} \tag{22}$$

with

$$-\left[\alpha_{ij}^{\langle a\rangle}\right] = \begin{bmatrix} -\frac{\theta}{2\pi} - \frac{\sin 2\theta_2 - \sin 2\theta_1}{2\pi(3-4\bar{\nu})} & \frac{\cos 2\theta_2 - \cos 2\theta_1}{2\pi(3-4\bar{\nu})} \\ \frac{\cos 2\theta_2 - \cos 2\theta_1}{2\pi(3-4\bar{\nu})} & -\frac{\theta}{2\pi} + \frac{\sin 2\theta_2 - \sin 2\theta_1}{2\pi(3-4\bar{\nu})} \end{bmatrix} \tag{23}$$

Here $\theta_2 - \theta_1 = \theta$ is the internal boundary angle subtended by material at $\mathbf{y}^{\langle a\rangle}$ on the boundary, as shown in Figure 2.3. Two special cases are worth noting. For an interior node, $\theta = 2\pi$; while for a boundary node where the boundary is smooth, $\theta = \pi$.

Substituting the above expressions into (20) leads to:

$$\alpha_{ij}^{\langle a\rangle}u_j^{\langle a\rangle} = \int_{\Gamma_u^{\langle a\rangle}+\Gamma_t^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)t_j(\mathbf{x})\,d\Gamma - \int_{C^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)u_j(\mathbf{x})\,d\Gamma$$
$$- \int_{\Gamma_u^{\langle a\rangle}+\Gamma_t^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left(u_j(\mathbf{x}) - u_j^{\langle a\rangle}\right)d\Gamma + \int_{\Omega^{\langle a\rangle}} b_j(\mathbf{x})\,\tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)d\Omega \qquad (24)$$

Equation (24) is the regularized local boundary integral equation. In (24), the subtraction method has been used in the third term of the right hand side. When the field point $\mathbf{x}$ approaches the source node $\mathbf{y}^{\langle a\rangle}$, $u_i(\mathbf{x}) - u_i^{<a>}$ tends to zero which removes the strong singularity and makes the integral numerically integrable. This eliminates the need to directly evaluate the CPV integral, and enhances the accuracy and efficiency of the model. All other terms in (24) are regular or weakly singular for which special integration quadrature gives convergent and accurate results. The regularized equation (24) is used for the current meshless implementation.

Equation (24) holds for any source node $\mathbf{y}^{\langle a\rangle}$, either inside the domain or on the boundary. The general strategy to solve the integral equation numerically is to select a number of source nodes and apply Equation (24) to each of these source nodes to obtain an equal number of governing equations. Some kind of interpolation or approximation is necessary to express the solution field in terms of nodal values at the source nodes. Numerical quadrature is then applied to evaluate the integrals, and the procedure converts the integral equations into a system of simultaneous algebraic equations. If all source nodes are selected on the boundary, a (regularized) boundary element method will result. In the current meshless integral method, both boundary and interior source nodes are used, and the moving least-squares approximation is employed for approximating the solution field, as presented in the next section.

In this work, attention is focused on 2D problems only where 1/r singularity has been regularized. For 3D cases, following similar derivation procedure an integral equation of the form of (17) can be obtained, with $C^{<a>}$ now representing the volume of a sphere center at node <a>, and $\Gamma^{<a>}$ representing the surface of that sphere. The order of singularity in $\tilde{t}_{ij}^*$ becomes $1/r^2$. With the application of the subtraction technique, an equation equivalent to (24) is obtained, and the order of singularity in the integrand $\tilde{t}_{ij}^*(u_j - u_i^{<a>})$ will be 1/r. However, it is known that 1/r singularity can be removed easily in a surface integral (see, for example, [39]).

The subtraction method can therefore be applied to regularize the boundary integral equation in 3D problems as well.

2.6 The Moving Least-squares Approximation

In the finite element method, the coupling between the nodes is accomplished through the use of shape functions, defined locally over each element, which interpolate the solution field from nodal values. For a meshless method, the absence of elements excludes the use of such shape functions and, therefore, a different local approximation scheme based on nodal values but independent of any elements needs to be devised. In this work, we have chosen to exploit the non-interpolative moving least-squares approximation (MLSA) scheme because of its high accuracy and the ease with which it can be extended to n-dimensional problems.

Consider a domain Ω that contains n nodes:

$$\mathbf{y}^{\langle a\rangle} = \begin{cases} \left[y_1^{\langle a\rangle}, y_2^{\langle a\rangle} \right]^T & \text{in 2-D} \\ \left[y_1^{\langle a\rangle}, y_2^{\langle a\rangle}, y_3^{\langle a\rangle} \right]^T & \text{in 3-D} \end{cases}, a = 1 \ldots n \tag{25}$$

For any of these nodes $\mathbf{y}^{\langle a\rangle}$, following [5][15][16], we define a ***support domain*** for node $\mathbf{y}^{\langle a\rangle}$, which is a sphere (3D) or disk (2D) centered on $\mathbf{y}^{\langle a\rangle}$ with a radius $\ell_w^{<a>}$. A weight function $w^{\langle a\rangle}$ is a continuous function that is positive in the support domain and zero outside, i.e.

$$\begin{cases} w^{\langle a\rangle}(\mathbf{x}) \geq 0 & \text{if } \left|\mathbf{x} - \mathbf{y}^{\langle a\rangle}\right| \leq \ell_w^{\langle a\rangle} \\ w^{\langle a\rangle}(\mathbf{x}) = 0 & \text{if } \left|\mathbf{x} - \mathbf{y}^{\langle a\rangle}\right| > \ell_w^{\langle a\rangle} \end{cases} \tag{26}$$

As introduced previously, the ***sub-domain*** $\Omega^{\langle a\rangle}$ for node $\mathbf{y}^{\langle a\rangle}$, located entirely inside Ω, is a sphere or part of a sphere centered on $\mathbf{y}^{\langle a\rangle}$ with a radius $h_S^{\langle a\rangle}$. Figure 2.4 illustrates the meaning of local sub-domain and support domain.

Two other frequently used concepts are the domain of definition and the domain of influence. The ***domain of definition*** of a point $\mathbf{x}$ is a set of all nodes whose weight functions are non-zero at $\mathbf{x}$, while the ***domain of influence*** of a node $\mathbf{y}^{<a>}$ is a set of all nodes whose weight functions are non-zero in some part or all of the sub-domain of node $\mathbf{y}^{<a>}$. The domain of

definition and the domain of influence are convenient terms in the description of MLSA and local boundary integrals, and are illustrated schematically in Figure 2.5.

The moving least-squares ***approximant*** $\mathbf{u}_h$ to a function $\mathbf{u}$ is defined by:

$$\mathbf{u_h}(\mathbf{x}) = \mathbf{p}^{\mathbf{T}}(\mathbf{x})\mathbf{c}(\mathbf{x}) \quad , \forall \mathbf{x} \in \Omega \tag{27}$$

The two vectors $\mathbf{p}$ and $\mathbf{c}$ are both functions of the spatial coordinates: $\mathbf{x} = [x_1, x_2]^T$ in 2D or $\mathbf{x} = [x_1, x_2, x_3]^T$ in 3D. $\mathbf{p}$ is a complete monomial basis of *m* terms (e.g., in 2D, m=3 for a linear basis, and 6 for a quadratic basis), $\boldsymbol{c}$ is a coefficient vector which is determined by minimizing a weighted discrete L_2-norm:

$$J(\mathbf{x}) = \sum_{a=1}^{Nx} \left(w^{\langle a \rangle}(\mathbf{x}) \left[\mathbf{p}^{\mathbf{T}}\left(\mathbf{y}^{\langle a \rangle}\right)\mathbf{c}(\mathbf{x}) - \hat{\mathbf{u}}^{\langle a \rangle} \right]^2 \right) \tag{28}$$

where $\hat{\mathbf{u}}^{\langle a \rangle}$ is the fictitious nodal displacement that approximates the value of $\mathbf{u}$ at node $\mathbf{y}^{\langle a \rangle}$, and the upper limit of summation, N_x, is the total number of nodes in the domain of definition of point $\mathbf{x}$. The matrices $\mathbf{P}$, $\mathbf{W}$ and $\hat{\mathbf{u}}$ are defined by

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^{\mathbf{T}}(\mathbf{y}^{<1>}) \\ \mathbf{p}^{\mathbf{T}}(\mathbf{y}^{<2>}) \\ \cdots \\ \mathbf{p}^{\mathbf{T}}(\mathbf{y}^{<N_x>}) \end{bmatrix}_{N_x \times m} \tag{29}$$

$$\mathbf{W}(\mathbf{x}) = \begin{bmatrix} w^{<1>}(\mathbf{x}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w^{<N_x>}(\mathbf{x}) \end{bmatrix}_{N_x \times N_x} \tag{30}$$

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}^{<1>} \\ \hat{\mathbf{u}}^{<2>} \\ \vdots \\ \hat{\mathbf{u}}^{<N_x>} \end{bmatrix} \tag{31}$$

Minimization of (28) leads to:

$$\mathbf{u_h}(\mathbf{x}) = \Phi^{\mathbf{T}}(\mathbf{x})\hat{\mathbf{u}} = \sum_{\mathbf{a}=1}^{Nx} \boldsymbol{\varphi}^{\langle a \rangle}(\mathbf{x})\hat{\mathbf{u}}^{\langle a \rangle} \tag{32}$$

where:

$$\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}) \tag{33}$$

$$\text{or } \boldsymbol{\varphi}^{\langle a\rangle}(\mathbf{x}) = \sum_{b=1}^{Nx} p_b(\mathbf{x})\left[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\right]_{ba} \tag{34}$$

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^{\mathbf{T}}\mathbf{W}(\mathbf{x})\mathbf{P} = \mathbf{B}(\mathbf{x})\mathbf{P} = \sum_{a=1}^{Nx} w^{\langle a\rangle}(\mathbf{x})\mathbf{p}\left(\mathbf{y}^{\langle a\rangle}\right)\mathbf{p}^{\mathbf{T}}\left(\mathbf{y}^{\langle a\rangle}\right) \tag{35}$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{P}^{\mathbf{T}}\mathbf{W}(\mathbf{x}) = \left[w^{\langle 1\rangle}(\mathbf{x})\mathbf{p}\left(\mathbf{y}^{\langle 1\rangle}\right) \quad w^{\langle 2\rangle}(\mathbf{x})\mathbf{p}\left(\mathbf{y}^{\langle 2\rangle}\right) \quad \cdots \quad w^{\langle Nx\rangle}(\mathbf{x})\mathbf{p}\left(\mathbf{y}^{\langle Nx\rangle}\right)\right] \tag{36}$$

The MLSA for a function exists only when $\mathbf{A}(\mathbf{x})$ is non-singular. A necessary condition for a well-defined MLSA is that for each sample point $\mathbf{x} \in \Omega$ (a node or a quadrature point), at least *m* weight functions are non-zero and the nodes in the domain of definition of $\mathbf{x}$ are not arranged in a degenerate pattern (such as on a straight line).

In MLSA, the shape function related to node $\mathbf{y}^{\langle a\rangle}$ is $\boldsymbol{\varphi}^{\langle a\rangle}(\mathbf{x})$. Since the shape function $\boldsymbol{\varphi}^{\langle a\rangle}$ is zero when the weight function $w^{\langle a\rangle}$ is zero, the interaction between nodal DOFs occurs only for nodes whose support domains are non-zero at node $\mathbf{y}^{\langle a\rangle}$. This means that the size of the support domain should be large enough to ensure the coupling between a minimum set of nodes, but small enough to capture local variations.

The smoothness of the basis functions and of the weight functions determines the smoothness of the shape function. Let C^k designate the space of k-*th* continuously differentiable functions. If $p_j \in C^h, \forall j \in \{1, 2, \cdots, m\}$ and $w^{\langle a\rangle} \in C^k, \forall a \in \{1, 2, \cdots, N_x\}$, then $\phi^{\langle a\rangle} \in C^r, \forall a \in \{1, 2, \cdots, N_x\}$ with $r = \min(k, h)$ [16].

The choices of the basis functions and the weight functions are very important and directly influence the behavior and the quality of the shape function. In this work, we use linear, quadratic and cubic monomial basis, and the Gaussian, spline and cosine weight functions, defined as follows:

Gaussian:
$$w^{\langle a\rangle}(\mathbf{x})=\begin{cases}\dfrac{\exp\left(-\left(\dfrac{r^{\langle a\rangle}}{s^{\langle a\rangle}}\right)^{2k}\right)-\exp\left(-\left(\dfrac{\ell_w^{\langle a\rangle}}{s^{\langle a\rangle}}\right)^{2k}\right)}{1-\exp\left(-\left(\dfrac{\ell_w^{\langle a\rangle}}{s^{\langle a\rangle}}\right)^{2k}\right)}, & 0\le r^{\langle a\rangle}\le \ell_w^{\langle a\rangle}\\ 0, & r^{\langle a\rangle}>\ell_w^{\langle a\rangle}\end{cases} \tag{37}$$

Spline:
$$w^{\langle a\rangle}(\mathbf{x})=\begin{cases}1-6\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^2+8\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^3-3\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^4, & 0\le r^{\langle a\rangle}\le \ell_w^{\langle a\rangle}\\ 0, & r^{\langle a\rangle}>\ell_w^{\langle a\rangle}\end{cases} \tag{38}$$

Cosine:
$$w^{\langle a\rangle}(\mathbf{x})=\begin{cases}\dfrac{1}{2}\left(1+\cos\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\pi\right)\right), & 0\le r^{\langle a\rangle}\le \ell_w^{\langle a\rangle}\\ 0, & r^{\langle a\rangle}>\ell_w^{\langle a\rangle}\end{cases} \tag{39}$$

Here $r^{<a>}$ is the distance from node $\mathbf{y}^{\langle a\rangle}$ to point $\mathbf{x}$, $\ell_w^{\langle a\rangle}$ is the size of support domain, $s^{<a>}$ is a scaling factor controlling the shape of the Gaussian weight function, and constant k is consistently taken to be 1 in our calculations. The coefficient $s^{<a>}$ has an important influence on the accuracy of the results, and a method to determine this constant has been suggested by Belytschko et al. [5]. The spline and cosine weight functions have only one parameter, the size of support $\ell_w^{\langle a\rangle}$, which makes their use simpler. In some examples, all weight functions were tested for comparison. It is noted that MLSA is non-interpolative, and there is a difference between the nodal value of the MLSA approximant $\mathbf{u}_h$ and the fictitious nodal displacement $\hat{\mathbf{u}}^{\langle a\rangle}$. For brevity, the subscript h in $\mathbf{u}_h$ will be omitted in the remaining part of this paper.

2.7 Meshless Implementation

We now apply the MLSA to the integral equation (24) to establish the meshless implementation. The shape function, as we have defined it, gives:

$$u_j(\mathbf{x})=\sum_{b=1}^{Nx}\varphi^{\langle b\rangle}(\mathbf{x})\hat{u}_j^{\langle b\rangle} \tag{40}$$

$$u_{j,k}(\mathbf{x}) = \sum_{b=1}^{Nx} \varphi_{,k}^{\langle b \rangle}(\mathbf{x})\, \hat{u}_j^{\langle b \rangle} \tag{41}$$

where N_x is the total number of nodes in the domain of definition of point $\mathbf{x}$.

The related traction term $t_j(\mathbf{x})$ is

$$t_j(\mathbf{x}) = \sigma_{ij}(\mathbf{x})\, n_i(\mathbf{x}) \tag{42}$$

where (n_1, n_2) is the normal to the plane passing $\mathbf{x}$ over which the traction acts. For a node $\mathbf{y}^{\langle b \rangle}$, we define $\mathbf{N}$ and $\mathbf{B}^{\langle b \rangle}$ matrices as:

$$N = \begin{bmatrix} n_1 & 0 & n_2 \\ 0 & n_2 & n_1 \end{bmatrix};\ B^{\langle b \rangle} = \begin{bmatrix} \varphi_{,1}^{\langle b \rangle} & 0 \\ 0 & \varphi_{,2}^{\langle b \rangle} \\ \varphi_{,2}^{\langle b \rangle} & \varphi_{,1}^{\langle b \rangle} \end{bmatrix} \tag{43}$$

Combining (43) with (41) and (42), we can express the traction in terms of the shape functions as follows:

$$\begin{Bmatrix} t_1(\mathbf{x}) \\ t_2(\mathbf{x}) \end{Bmatrix} = \sum_{b=1}^{Nx} \left[NCB^{\langle b \rangle}\left(\mathbf{y}^{\langle a \rangle}\right) \right] \begin{Bmatrix} \hat{u}_1^{\langle b \rangle} \\ \hat{u}_2^{\langle b \rangle} \end{Bmatrix} \tag{44}$$

here $\mathbf{C}$ is the stiffness matrix (which works for both plane strain and plane stress):

$$C = \begin{bmatrix} \dfrac{(1-\bar{\nu})\bar{E}}{(1+\bar{\nu})(1-2\bar{\nu})} & \dfrac{\bar{\nu}\bar{E}}{(1+\bar{\nu})(1-2\bar{\nu})} & 0 \\ \dfrac{\bar{\nu}\bar{E}}{(1+\bar{\nu})(1-2\bar{\nu})} & \dfrac{(1-\bar{\nu})\bar{E}}{(1+\bar{\nu})(1-2\bar{\nu})} & 0 \\ 0 & 0 & \dfrac{\bar{E}}{2(1+\bar{\nu})} \end{bmatrix} \tag{45}$$

With the above discretization and boundary conditions that $u_j = \bar{u}_j$ on $\Gamma_u^{<a>}$ and $t_j = \bar{t}_j$ on $\Gamma_t^{<a>}$, equation (24) becomes (there is a summation on b and j but not on a and i):

$$\sum_{b=1}^{N_y} \left(H_{ij}^{<a,b>} \hat{u}_j^{<b>} \right) - \sum_{b=1}^{N_y} \left(L_{ij}^{<a,b>} \hat{u}_j^{<b>} \right) + \alpha_{ij}^{<a>} u_j^{<a>} = G_i^{<a>} \tag{46}$$

where $H_{ij}^{\langle a,b \rangle}$, $L_{ij}^{<a,b>}$ and $G_i^{\langle a \rangle}$ are:

$$H_{ij}^{\langle a,b\rangle} = -\int_{\Gamma_u^{\langle a\rangle}} \tilde{u}_{ik}^{*}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left[\mathbf{NCB}^{\langle b\rangle}\left(\mathbf{x}\right)\right]_{kj} d\Gamma + \int_{C^{\langle a\rangle}} \tilde{t}_{ij}^{*}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\phi^{\langle b\rangle}\left(\mathbf{x}\right) d\Gamma + \int_{\Gamma_t^{\langle a\rangle}} \tilde{t}_{ij}^{*}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left(\phi^{\langle b\rangle}\left(\mathbf{x}\right)\right) d\Gamma \tag{47}$$

$$L_{ij}^{<a,b>} = \int_{\Gamma_t^{<a>}} \tilde{t}_{ij}^{*}(\mathbf{x},\mathbf{y}^{<a>})(\phi^{<b>}(\mathbf{y}^{<a>}))d\Gamma \tag{48}$$

$$G_i^{\langle a\rangle} = \int_{\Omega^{\langle a\rangle}} b_j\left(\mathbf{x}\right)\tilde{u}_{ij}^{*}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right) d\Omega + \int_{\Gamma_t^{\langle a\rangle}} \tilde{u}_{ij}^{*}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\overline{t}_j\left(\mathbf{x}\right) d\Gamma - \int_{\Gamma_u^{\langle a\rangle}} \tilde{t}_{ij}^{*}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left(\overline{u}_j\left(\mathbf{x}\right) - \overline{u}_j\left(\mathbf{y}^{\langle a\rangle}\right)\right) d\Gamma \tag{49}$$

and the upper limit of summation, N_y, is the total number of nodes in the domain of influence of node $\mathbf{y}^{<a>}$.

In all numerical examples tested in this work, the body force term is zero.

2.8 Treatment for Weak Singularity

The second integral in the right hand side of Equation (47) of $H_{ij}^{<a,b>}$ and the first integral in the right hand side of Equation (49) are regular for which standard quadrature can be used with good accuracy. Owing to the subtraction technique, the singularity in the third integral in (47) as **x** approaches $\mathbf{y}^{<a>}$ is cancelled by the term in (48), and similarly the third integral in (49) is also regularized.

Even though the subtraction technique removes the strong singularity, the integrands in the first integral of (47) and the second integral in (49) still contain the weakly singular ln(r) term. The logarithmic singularity is integrable, but the accuracy of ordinary Legendre-Gauss integration is poor. We found that the special integration scheme for the logarithmic singularity $\int_0^1 f(r)\ln\left(\frac{1}{r}\right)dr = \sum_{i=1}^{N} w_i f(r_i)$ in [36], which is reproduced in Table 2.1 for completeness, achieves excellent numerical accuracy. As will be discussed in the Results section, by using the special integration the accuracy of the convergence patch tests under constant strain is within machine accuracy.

In our numerical examples, the numbers of integration points were as follows: 8 integration points for any integral along a straight line, and 64 integration points for any integral over an interior sub-domain. For a boundary node for which the local sub-domain is a partial circle, we just computed the nominal 64 integration points and then removed all the nodes located outside the domain. For regularized integrals, the usual Legendre-Gauss integration was used. For integrals containing logarithmic singularity, the special integration of 8 integration points listed in Table 2.1 was used.

2.9 Imposition of Essential and Natural Boundary Conditions

Appropriate boundary conditions need to be imposed in order to solve the simultaneous equations (46). In meshless methods, imposing the essential (Dirichlet) boundary conditions is not as trivial as in the finite element method. Because MLSA is non-interpolative, the essential boundary condition does not take the form of prescribed value for the fictitious nodal displacement ($\hat{u}_i^{<a>} = \bar{u}_i^{<a>}$), but rather a constraint equation involving a linear combination of the fictitious nodal displacements in a neighborhood of the boundary node (i.e., $u_i^{\langle a\rangle} = \sum_{b=1}^{n} \phi^{\langle b\rangle}\left(\mathbf{y}^{\langle a\rangle}\right)\hat{u}_i^{\langle b\rangle} = \bar{u}_i^{\langle a\rangle}$). A number of techniques for the imposition of essential boundary conditions have been developed, including: (1) Collocation methods [6][49][50][51][52]; (2) Lagrange multiplier method [5][10][7][53][54]; (3) Penalty method [55][50][31]; (4) Nitsche's method [56][57]; (5) Coupled meshless-finite element method [58][59][60][61]; (6) Admissible approximation method [62]; (7) Method based on d'Alembert principle [63]; (8) Use of window or correction functions that vanish on the boundary [13]; (9) Discrete form of essential boundary conditions [8]; and (10) Displacement constraint equation method [64].

The list is intended to illustrate the various methods that have been proposed in the literature, and is not an exhaustive list. All methods have advantages and disadvantages. The Lagrange multiplier method [5][10][7][53][54] is a general method to enforce constraints (linear or nonlinear) in a large variety of problems; its implementation is straightforward, and the boundary conditions are imposed rather accurately. The method, however, introduces a new unknown function, the Lagrange multiplier, and leads to an increased problem size and a poorly conditioned matrix equation. The penalty method [55][50][31] does not increase the problem size, requires only the choice of one scalar parameter, and is easier to implement than the

Lagrange method. However, large values of the penalty parameter must be used which may lead to ill-conditioning of the system equations, and the boundary conditions are enforced only approximately. The Nitsche's method [56][57], based on a modified weak form that involves a positive constant scalar parameter, does not increase the problem size, and does not suffer ill-conditioning. In practice, the value of the scalar parameter need not be very large. The implementation of the Nitsche's method, however, is not as general and straightforward as for the Lagrange multiplier method or the penalty method because the modification of the weak form in the Nitsche's method is different for each particular problem. The essential boundary conditions in Nitsche's method are also satisfied approximately. Coupled meshless-finite element method [58][59][60][61] divides the problem domain into an element interpolation zone where essential boundary conditions are to be imposed, a meshless approximation zone, and a transition zone where finite element interpolation and meshless approximation are blended continuously from one type to the other. Since the FEM shape function at the essential boundary nodes is in fact interpolative and satisfies the Kronecker delta property (i.e., the shape function for node b at node a satisfies $\phi^{<b>}(\mathbf{y}^{<a>}) = \delta_{ab}$), essential boundary conditions can be imposed directly and accurately as has been done routinely in FEM. Naturally, the method loses the advantages the meshless method offers in the boundary element zone. It also complicates programming because of the need to handle additionally the FE interpolation and the blending of meshless approximation and FE interpolation in the transition zone. The d'Alembert's principle has been used for general constraints in both meshless methods and FEM in which n differential equations and m constraints are replaced by n-m equations through choosing n-m generalized, independent variables [63]. The method reduces the problem size, but it requires the orthogonalization of a large square matrix with a size equal to the total number of nodes. In the admissible approximation approach [62], the essential boundary conditions are imposed by forcing the weight function to be zero on essential boundaries. In the discrete form of essential boundary conditions, the trial function is modified to satisfy the essential boundary conditions based on the weak form. It is shown in [8] that the weak form of essential boundary conditions is identical to the Lagrange multiplier method if the same shape functions are used for the Lagrange multipliers and the test and trial functions. In the displacement constraint equations method [64], the essential boundary conditions are treated as displacement constraints, and the fictitious nodal displacements are partitioned into two subvectors: (1) constrained subvector corresponding to the

essential boundary nodes, and (2) unconstrained subvector corresponding to the remaining nodes. All other matrices and vectors are partitioned in the same way. The partition enables the constrained subvector to be expressed in terms of a linear combination of the prescribed displacement values on the essential boundary and the unconstrained subvector. The resultant system equations therefore absorb the essential boundary conditions, and the stiffness matrix is symmetric, positive, and banded.

Collocation methods for enforcing essential boundary conditions are defined as those methods in which conditions are enforced exactly at a discrete set of boundary nodes [51]. This is accomplished by replacing rows of the matrix equations obtained from the appropriately discretized weak form with equations that ensure the enforcement of the essential boundary conditions. A number of collocation methods have been developed. The direct collocation method [6] used the collocation condition

$$\hat{u}_i^{<a>} = \overline{u}_i^{<a>} \tag{50}$$

This is actually inconsistent with the assumption of MLSA since the fictitious nodal displacement $\hat{u}_i^{<a>}$ is generally not equal to the approximated displacement value. The same collocation condition was used in [49] to enforce boundary condition with a new definition of the MLSA L_2 norm which was shown to improve the solution accuracy.

A modified collocation method uses

$$u_i^{\langle a\rangle} = \sum_{b=1}^{n} \phi^{\langle b\rangle}\left(\mathbf{y}^{\langle a\rangle}\right)\hat{u}_i^{\langle b\rangle} = \overline{u}_i^{\langle a\rangle} \tag{51}$$

as the collocation condition which was shown to yield more accurate results [50]. Wagner and Liu [51] pointed out that when the shape functions do not satisfy the Kronecker delta property, the rows of the matrix equation corresponding to the essential boundary nodes contribute to the solution of the displacement field, and that when these equations are simply ignored, as is done in the traditional (including direct and modified) collocation method, the weak form is not satisfied consistently. Such inconsistency leads to poor accuracy and low convergence rate. They developed a corrected collocation method which restores the consistency of the weak form and enhances convergence. Wu and Plesha [52] proposed a boundary flux collocation method to enforce the boundary conditions exactly which maintains the consistency of the weak form and

avoids partitioning and rearrangement of the nodal DOFs into a constrained and a free group as needed in the corrected collocation method.

Generally, there are two types of discretization in meshless methods: (1) collocation over multiple local domains, which is used in SPH [3], FPM [65], and meshless method based on LBIE [15][16][17][18], etc.; (2) Galerkin based method over the global domain, which is used in EFG [5][6][7][8], clouds [13], RKPM [9][10][11][12], etc. For collocation based discretization, each equation is obtained by applying the weak form over a particular local domain, and the weak form needs to be applied n times for a problem with n DOFs. Consequently, traditional collocation method with (51) can be used easily and directly to impose essential boundary conditions, because each of the system equations is independent of the rest, and replacing the equation corresponding to a constrained DOF by (51) will not cause any inconsistency in the weak form. For the Galerkin based discretization, the n system equations are obtained by applying the weak form over the global domain once, and therefore all equations must hold simultaneously in order to maintain consistency in the weak form. Replacing a row in the matrix equation by (51), which contains a linear combination of DOFs rather than dictating the value of the single constrained DOF, sacrifices the consistency of the weak form and compromises the accuracy of the solution.

The current meshless integral method utilizes the collocation based discretization (over multiple sub-domains), and the traditional collocation method can be directly used to impose essential boundary conditions. Since the system equations are obtained by applying the integral Equation (24) to each source node over a local sub-domain, for a DOF with essential boundary condition, we simply use the essential boundary condition $u_i = \bar{u}_i$ rather than applying the integral Equation (24). This is equivalent to replacing the governing equation corresponding to the DOF with essential boundary condition (51). Our numerical tests presented in the next section show that the tradition collocation method for imposing essential boundary conditions works very well with the meshless integral method.

For the natural boundary condition $t_i = \bar{t}_i$, no special treatment is needed. The prescribed traction is directly used in the second integral in Equation (49).

After the boundary conditions are imposed, the governing equations can be written as

$$\sum_{b=1}^{N_y} (K_{ij}^{<a,b>} \hat{u}_j^{<b>}) = F_i^{<a>} \tag{52}$$

where

$$K_{ij}^{<a,b>} = \begin{cases} H_{ij}^{<a,b>} - L_{ij}^{<a,b>} + \alpha_{ij}^{<a>}\phi^{<b>}(\mathbf{y}^{<a>}) & \text{when } u_i^{<a>} \text{ is unconstrained} \\ \phi^{<b>}(\mathbf{y}^{<a>})\delta_{ij} & \text{when } u_i^{<a>} = \bar{u}_i^{<a>} \end{cases} \tag{53}$$

$$F_i^{<a>} = \begin{cases} G_i^{<a>} & \text{when } u_i^{<a>} \text{ is unconstrained} \\ \bar{u}_i^{<a>} & \text{when } u_i^{<a>} = \bar{u}_i^{<a>} \end{cases} \tag{54}$$

and the upper limit of summation, N_y, is the total number of nodes in the domain of influence of node $\mathbf{y}^{<a>}$.

2.10 Numerical Examples

This section presents the numerical solutions to several examples using the meshless integral method developed in this study and uses analytical solutions given in the literature as the bases for comparison. The tests include the lower and higher order convergence patch tests, a cantilever beam and a hole in an infinite plate. The density and distribution of the nodes were varied to assess their effects on the solution accuracy. The effects of the weight functions and monomial basis were also investigated. There are three geometric parameters (only two if spline or cosine weight function is used) for each node: the size of sub-domain $h_s^{\langle a \rangle}$, the size of support domain (or the size of weight function) $\ell_w^{<a>}$, and the scaling factor $s^{\langle a \rangle}$ for the Gaussian weight function. It is difficult to derive theoretical formulas for determining the optimal values of the geometric parameters $h_s^{\langle a \rangle}$ and $\ell_w^{<a>}$. The following guidelines are recommended for the selection of these parameters.

The sub-domain radius $h_s^{\langle a \rangle}$ is selected in order to simplify the domain of integration, $C^{<a>}$. For an interior node, the sub-domain should be fully enclosed by the problem domain so that the sub-domain boundary is a full circle; for a boundary node, the sub-domain should not contain any boundary corner so that the sub-domain boundary closely resembles a half circle; for a boundary corner node, the sub-domain closely resembles a circular fan. It is worth noting that for a node $\mathbf{y}^{\langle a \rangle}$, while the support domains of the neighboring nodes affect the invertibility of the **A** matrix in MLSA at $\mathbf{y}^{\langle a \rangle}$, the size of the sub-domain $h_s^{\langle a \rangle}$ does not.

The size of the support domain $\ell_w^{<a>}$ has a stronger influence on the accuracy of the model than $h_s^{\langle a\rangle}$ does. The choice of $\ell_w^{<a>}$ has to satisfy several conditions. First, the union of all support domains must be equal to or larger than the union of all sub-domains in order for every integral in the governing equations to have definition. Second, the union of all support domains must be equal to or larger than the problem domain in order for the solution field to have definition over the whole problem domain. Third, it should be chosen such that the matrix $\mathbf{A}$ in MLSA Equation (33) is not singular. This implies that $\ell_w^{<a>}$ should be large enough so that for any given point **x**, there are at least m nodes whose weight functions are non-zero at **x** if an m-term monomial basis is used.

Manually setting up the sub-domain radius and support domain radius for each node can be rather time consuming. In this work, we have developed algorithms for determining the two parameters for each node automatically. For the sub-domain radius $h_s^{\langle a\rangle}$, the distance between node a and the nearest neighboring node and the shortest distance from node a to the problem domain boundary are calculated, and the smaller of the two is taken as $h_s^{\langle a\rangle}$. For support domain radius $\ell_w^{<a>}$, $h_s^{<a>}$ times a constant C of the order 1 is taken as the value for $\ell_w^{<a>}$. The constant C is adjusted so that the number of nodes inside the domain of definition of each node is equal to the minimum required by MLSA to ensure a non-singular $\mathbf{A}(\mathbf{x})$ plus a small number (usually 5 to 7). The algorithm, carried out during program execution, frees the users from the time consuming task of setting up the two geometric parameters manually during input.

A global error indicator, the L_2-norm error in displacement, is defined by

$$L_2\text{-norm error} = \frac{\left\{\sum_{j=1}^{N}\left\{\left(u_1^{\langle j\rangle}\Big|^{num} - u_1^{\langle j\rangle}\Big|^{exact}\right)^2 + \left(u_2^{\langle j\rangle}\Big|^{num} - u_2^{\langle j\rangle}\Big|^{exact}\right)^2\right\}\right\}^{1/2}}{\left\{\sum_{j=1}^{N}\left\{\left(u_1^{\langle j\rangle}\Big|^{exact}\right)^2 + \left(u_2^{\langle j\rangle}\Big|^{exact}\right)^2\right\}\right\}^{1/2}} \times 100\% \qquad (55)$$

The L_2-norm error is a better measure of the overall performance of a numerical model than the conventional relative error for a single point. In this work, we will use the global L_2-norm error as a measure of the overall performance of the numerical method.

2.10.1 Constant Stress Patch Tests

The first patch test is a 2×2 m^2 square plate under linear displacement field or constant traction on the four edges. These patch tests have been widely used in previous work [5][16][66] for testing the validity of a numerical method. The material properties are $E = 1000$ MPa, $\nu = 0.3$ with plane strain condition. Three meshless models, which are shown in Figure 2.6, were simulated. The first is with 9 regular nodes, the second with 25 regular nodes, and the third with 25 irregular nodes. This was done to test the effects of model refinement and distribution of nodes. Since the exact solution is linear in displacements, we used only the linear basis for MLSA.

For all three patches, two types of boundary conditions, case 1 and case 2, both corresponding to the same physical problem, were tested. For case 1, prescribed displacements are applied to all edges of the plate. For case 2, prescribed displacements are applied to the left and bottom edges, while prescribed tractions are applied to the top and right edges. The spline weight function was used in these tests. The L_2-norm errors for all models are reported in Table 2.2. For case 1, the L_2-norm errors in displacement for the 9 node, 25 node regular and 25 node irregular models are 2.1×10^{-16}, 4.6×10^{-16} and 9.2×10^{-15} respectively. For case 2, the corresponding L_2-norm errors are 6.7×10^{-11}, 1.5×10^{-10} and 3.6×10^{-10} respectively. The results indicate that the method successfully passes all three patch tests within machine accuracy.

2.10.2 Higher-order Patch Tests

Two higher order patches under plane stress condition, as shown in Figure 2.7 and Figure 2.8, were tested. The first has 28 uniformly distributed nodes, and the second has 14 non-uniformly distributed nodes. The same patches have been studied in [5]. Two types of loading conditions were enforced. For the first case, the right edge is loaded with a uniform normal stress of 1000 MPa; the upper edge is traction free, while the bottom edge is constrained from moving in the x_2 direction and the left edge constrained from moving in the x_1 direction. For the second loading case, conditions for the top and left edges remain unchanged, while the right edge is subjected to a normal stress that varies linearly with x_2 from zero at the lower right corner to 3000 MPa at the upper right corner; and the bottom edge is constrained from moving in x_1

direction. Loading case 1 therefore corresponds to uni-axial tension, while loading case 2 corresponds to pure bending.

The material properties used in the meshless calculation are E=1000 MPa $\nu = 0.25$. The exact solutions are $u_1 = x_1$ and $u_2 = -\frac{x_2}{4}$ for case 1; $u_1 = x_1 x_2$ and $u_2 = -\frac{4x_1^2 + x_2^2}{8}$ for case 2. Based on the functional form of the exact solutions, a linear basis for the shape function was used in case 1 and a quadratic basis used in case 2.

The L_2-norm errors for all meshless models are reported in Table 2.3. For load case 1, the L_2-norm errors for these two models are 1.4×10^{-8} and 2.1×10^{-8} respectively; for load case 2, the L_2-norm errors are 1.0×10^{-7} and 1.1×10^{-7} respectively. These results show that the L_2-norm error for each test is roughly within machine accuracy and that the meshless integral method successfully passes these two high order patch tests.

2.10.3 Cantilever Beam

Next, we consider the cantilever beam as shown in Figure 2.9a. A closed form solution for plane stress condition has been given in [67]:

$$\begin{cases} \sigma_{11} = -\dfrac{P}{I} x_1 x_2 \\ \sigma_{22} = 0 \\ \sigma_{12} = -\dfrac{P}{2I}\left(1 - \dfrac{x_2^2}{c^2}\right) c^2 \end{cases} \tag{56}$$

and

$$\begin{cases} u_1 = -\dfrac{P x_1^2 x_2}{2EI} - \dfrac{\nu P x_2^3}{6EI} + \dfrac{P x_2^3}{6\mu I} + \left(\dfrac{PL^2}{2EI} - \dfrac{Pc^2}{2\mu I}\right) x_2 \\ u_2 = \dfrac{\nu P x_1 x_2^2}{2EI} + \dfrac{P x_1^3}{6EI} - \dfrac{PL^2 x_1}{2EI} + \dfrac{PL^3}{3EI} \end{cases} \tag{57}$$

Here c=0.5 m is half the height of the beam, $I = 2c^3/3$, L=8 m, P=1 N, $E = 1000$ MPa, $\nu = 0.3$, $\mu = E/(2(1+\nu))$. For the above solution to be exact, tractions according to Equation (56) or displacements according to Equation (57) are applied to the free end of the beam.

Meshless models with various densities of evenly distributed nodes were tested: a coarse model with 85 nodes (5 rows by 17 columns), a medium one with 165 nodes (5 rows by 33

columns) and a fine one with 585 nodes (9 rows by 65 columns). Figure 2.9b, Figure 2.9c and Figure 2.9d display the three models. Essential boundary conditions were applied on the left and right edges according to equation (56), and natural boundary conditions (traction free) were applied on the top and bottom edges. The effects of weight functions and the monomial basis were investigated.

Figure 2.10 displays the deformed mesh for the 85-node model using cubic basis and spline weight function. For this case, the L_2-norm error in displacement is 2.0×10^{-6}, and the numerical results (squares) virtually coincide with analytical solutions (triangle). For the 165-node model, the L_2-error is further reduced to 9.3×10^{-7}. The higher accuracy is expected because the exact solution is cubic in displacement. Table 2.4 summarizes the performance of the three meshless models.

Figure 2.11 compares the effects of three monomial bases (linear, quadratic and cubic) for the 85-node meshless model using the three weight functions. It is clear that the choice of the monomial basis has a stronger influence on the accuracy than the choice of the weight function. Regardless of the weight function used, the linear basis produces the largest error (of the order of 0.01). The quadratic basis improves the accuracy by roughly a factor of 10, while the cubic basis yields very accurate results, with errors of the order of 10^{-6} for all three weight functions. The performances of the three weights are comparable. In the remaining calculations, we used only the spline weight function because it performs slightly better than the cosine weight function, and requires fewer free geometric parameters than the Gaussian weight function does, which makes the simulation simpler.

Figures 2.12 and 2.13 compare the results using the original LBIE method [16] and the results using the current method. The relative errors (based on the analytical solution) of the displacement at the top of the free end (x, y)=(0.0, 0.5) for five cases are shown in Figure 2.12, and the relative errors of the shear stress at midpoint (x, y)=(4, 0) for the corresponding cases are shown in Figure 2.13. The improvement in accuracy of the current meshless integral method over the original method [16] is evident.

The distributions of the normal bending stress σ_{11} from both the meshless model and FEM (using ANSYS [68]) follow the analytical results quite accurately. Accurate distribution of the shear stress σ_{12} is usually more difficult to obtain from numerical models. It is therefore

more revealing to compare shear stress distributions. Figure 2.14 shows the effects of the monomial basis on the computed shear stress along the mid-vertical cross-section at $x_1 = 4$ for the 85-node model with the spline weight function. It is noteworthy that the results using the cubic basis, with L_2-norm error of 1.7×10^{-8}, practically coincide with the exact solutions. The quadratic basis also yields satisfactory results with a L_2-norm error of 0.0058. The linear basis produces larger errors with a L_2-norm error of 0.17, and the shape of the distribution is also erroneous.

It is revealing to compare the shear stress obtained from the meshless integral method with FEM results, as shown in Figure 2.15. The meshless results with cubic basis are practically indistinguishable from the analytic solution. For the FEM model, quadratic elements (quadrilateral 8-node element) are used and the total number of nodes is 95. The FEM curve for the shear stress is trapezoidal in shape. In contrast, the meshless results are smooth. It is worth noting that in FEM, since stresses and strains are discontinuous across element edges, it is a common practice to post-process the stresses and strains. During post processing, the stresses or strains are extrapolated from Gaussian points to nodes, and average values over all neighboring elements at each node are taken as the corresponding nodal stresses or strains. Such post-processing is unnecessary in the meshless integral method, because the stresses and strains are already smooth enough.

A new, truly element free postprocessor has been developed and is used for graphical representation of various results from the meshless integral method. This postprocessor will be introduced in detail in a forthcoming paper. Contour plots for σ_{11} and σ_{12} from the meshless model are presented in Figure 2.16 and Figure 2.17. They match well the analytical results shown in Figure 2.18. Comparison between the two sets of results reveals that the cubic basis performs best.

2.10.4 Infinite Plate with a Circular Hole

The last test example is an infinite plate with a circular hole in the center subjected to a uniform remote tension σ in x_1 direction. The exact solutions for the stresses are given by [67]:

$$\begin{cases} \sigma_{11} = \sigma\left[1 - \dfrac{a^2}{r^2}\left(\dfrac{3}{2}\cos 2\theta + \cos 4\theta\right) + \dfrac{3a^4}{2r^4}\cos 4\theta\right] \\ \sigma_{22} = \sigma\left[-\dfrac{a^2}{r^2}\left(\dfrac{1}{2}\cos 2\theta - \cos 4\theta\right) - \dfrac{3a^4}{2r^4}\cos 4\theta\right] \\ \sigma_{12} = \sigma\left[-\dfrac{a^2}{r^2}\left(\dfrac{1}{2}\sin 2\theta + \sin 4\theta\right) + \dfrac{3a^4}{2r^4}\sin 4\theta\right] \end{cases} \tag{58}$$

where (r, θ) are the polar coordinates with origin at the center of the hole, and θ is measured from the positive x_1 axis counterclockwise. The corresponding displacements are:

$$\begin{cases} u_1 = \dfrac{1+\bar{\nu}}{\bar{E}}\sigma\left((1-\bar{\nu})r\cos\theta + 2(1-\bar{\nu})\dfrac{a^2}{r}\cos\theta + \dfrac{1}{2}\dfrac{a^2}{r}\cos 3\theta - \dfrac{1}{2}\dfrac{a^4}{r^3}\cos 3\theta\right) \\ u_2 = \dfrac{1+\bar{\nu}}{\bar{E}}\sigma\left(-\bar{\nu}r\sin\theta - (1-2\bar{\nu})\dfrac{a^2}{r}\sin\theta + \dfrac{1}{2}\dfrac{a^2}{r}\sin 3\theta - \dfrac{1}{2}\dfrac{a^4}{r^3}\sin 3\theta\right) \end{cases} \tag{59}$$

Because of symmetry, only the upper right quadrant of the plate was modeled (see Figure 2.19). The actual model used in the simulation is a finite sized plate of 5 m by 5 m with a quarter of a circular hole of radius of 1 m centered at the lower left corner. The model geometry has 336 nodes with nodes concentrated around the hole. The material properties of E=1000 MPA and ν=0.3 and the remote traction $\sigma = 1$ MPA are used for Equations (58) and (59), and plane strain condition is enforced. Two types of boundary conditions are imposed on the finite sized model. For case 1, essential boundary conditions according to equation (59) were applied on the right, the upper, the left and the bottom edges. On the remaining edge (the inner circle), the natural boundary conditions (free traction) were applied. For case 2, essential boundary conditions according to equation (59) were applied on the left and the bottom edges, while on the remaining edges, natural boundary conditions according to equation (58) were applied.

The spline weight function was used in the calculation, and different $\ell_w^{<a>}$'s were assigned for different nodes. Generally, smaller $\ell_w^{<a>}$ is used for regions with denser nodal distribution. Since the exact displacement solution (equation (59)) is not polynomial in x_1 and x_2, and the use of higher order basis does not necessarily result in more accurate results, only linear and quadratic bases were tested. The L_2-norm error in displacement and the normal stress on the left edge are monitored and reported.

Figure 2.20 and Figure 2.21 show the deformed meshless model (squares), using spline weight function and linear basis, for case 1 and case 2 respectively. The meshless results are in good agreement with analytical results (triangles), and the L_2-norm errors are 0.0032 and 0.0033 respectively. The L_2-norm errors for the three meshless models (336 nodes, 1271 nodes, and 2806 nodes) under two boundary conditions and the corresponding results for FEM (ANSYS) models are listed in Table 2.5. From Table 2.5, for both cases the accuracy of the meshless model with linear basis improves with model refinement (i.e. increased nodal density). For case 1, the L_2-norm errors are 0.0032, 0.0022 and 0.0017 for the three models respectively, while for case 2, the L_2-norm errors are 0.0033, 0.0023 and 0.0018 for the three models respectively. The performance of the 336 node model is comparable with that of the FEM model with the same number of nodes. However, the performance of the 1271 node model and the 2806 model is not as good as that of the corresponding FEM models.

Figure 2.22 and Figure 2.23 present the distribution of normal stress σ_{11} along $x_1 = 0$ of the 336-node model for load case 1 and case 2 respectively. Figure 2.24 shows the distribution of normal stress σ_{11} along $x_1 = 0$ of the 1271-node model for load case 1. For all cases, the numerical results are in reasonable agreement with the analytic solution. Figure 2.25 shows the distribution of normal stress σ_{11} along $x_1 = 0$ of the 120-node model using the original LBIE method presented in [16] in which the support domain size is 3, and Gaussian support function is used with the scaling factor s of 0.54 for linear basis and 0.44 for quadratic basis. It should be noted that a direct, fair comparison between our method and the original LBIE [16] is impossible because there are differences in the nodal distribution, geometric parameters, number of numerical integration points, and more importantly, the form of the governing equation on which the method is based. The best we can do is to select models that are the closest from the available results. When comparing Figure 2.25 with Figure 2.22, we find that results from our method follow more closely the analytical solution than those from the original LBIE method [16].

Figures 2.26 and 2.27 show the contour plots of σ_{11} stress distributions from the meshless integral method for the linear and quadratic basis, while Figures 2.28 and 2.29 show the corresponding results from FEM for comparison. Figures 2.30 and 2.31 are the contours of σ_{12} stress distributions from the meshless, and Figures 2.32 and 2.33 are the corresponding results from FEM. Analytical results for σ_{11} and σ_{12} are presented in Figure 2.34 and Figure 2.35

respectively. A comparison of these figures reveals that results from the meshless integral method are closer to the analytical results than those from FEM. It also reveals that with the meshless integral method, the quadratic basis performs better than the linear basis for this problem.

2.11 Concluding Remarks

A meshless integral method based on regularized local boundary integral equation has been presented. The meshless integral method was built on an earlier method proposed by Atluri and coworkers [15][16] based on the local boundary integral equation representation for elasticity. The implementation introduced in the present work offers several key improvements: (1) the strong singularity (1/r type) in the governing equations has been removed. The formula is rather straightforward and more accurate than the direct limit approach used in [16], and simpler than the lengthy singularity removal treatment proposed in [34]. (2) A special numerical integration is employed for the calculation of integrals with weak singularity (logarithmic type) which further improves accuracy. (3) A treatment is proposed for handling the essential boundary conditions explicitly, which requires little or no additional computational cost and offers high accuracy. This is a great advantage considering that the enforcement of essential boundary conditions for some meshless methods can be quite cumbersome because of the non-interpolative moving least-squares approximation. (4) The natural boundary conditions are incorporated in the system governing equation and require no special handling.

The meshless integral method developed in this study has shown excellent accuracy in the lower and higher order patch tests and the cantilever beam problem, and satisfactory performance for the hole problem. The use of exact, singular kernels (the fundamental solutions) contributes to the high accuracy of the governing boundary integral equations, while the singularity removal scheme introduced in this work makes it possible to evaluate the integral equation numerically. The method can be extended to 3D linear elasticity without major difficulties. The advantages of the present meshless integral methods over other meshless methods published in the literature include the following: the essential boundary conditions can be enforced exactly and easily; no background mesh is needed to evaluate the domain and boundary integrals; since the governing boundary integral equations, based on the fundamental solution, are exact and approximation comes only from the moving least-squares approximation

and the numerical integration, the method can achieve high accuracy; the calculation of derived variables (such as stress fields) involves only a small regular sub-domain instead of the global domain/boundary; and the method offers flexibility in node adaptivity and model refinement.

2.12 Acknowledgement

The support from the U.S. Department of Energy (grant number: DE-FG02-00ER45845) and the Advanced Manufacturing Institute at Kansas State University is gratefully acknowledged.

2.13 References

[1] Bathe, K.J. *Finite Element Procedures in Engineering Analysis*, Prentice-Hall: Englewood Cliffs, 1982.

[2] Zienkiewicz, O.C. and Taylor, R.L. *The Finite Element Method*, McGraw-Hill: Berkshire, 1991.

[3] Gingold RA, Moraghan JJ, Smoothed particle hydrodynamics: theory and applications to non-spherical stars, Monthly Notices of the Astronomical Society 1977, 181, 375-389

[4] Nayroles, B., Touzot and G., Villon, P. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational Mechanics. 1992; 10: 307-318.

[5] Belytschko, T., Lu, Y.Y. and Gu, L. Element free Galerkin method. International Journal for Numerical Methods in Engineering. 1994; 37: 229-256.

[6] Lu, YY, Belytschko T, and Gu, L, A new implementation of the element free Galerkin method, Computer Methods in Applied Mechanics and Engineering, 1994, 113, pp.397-414

[7] Krysl P, Belytschko T., Analysis of thin plates by the element-free Galerkin method, Computational Mechanics, 1998, 21, 211-222

[8] Lu YY, Belytschko T, Tabbara M. Element-free Galerkin method for wave propagation and dynamic fracture, Computer Methods in Applied Mechanics and Engineering, 1995, 126, pp.131-153

[9] Liu, W.K. and Chen. Y. Wavelet and multiple scale reproducing kernel methods. International Journal for Numerical Methods in Fluids. 1995; 21: 901-931.

[10] Aluru, N.R. A reproducing kernel particle method for meshless analysis of microelectromechanical systems. Computational Mechanics. 1999; 23: 324-338

[11] Chen, J.S., Pan, C., and Wu, C.T. A Lagrangian reproducing kernel particle method for metal forming analysis. Computational Mechanics. 1998; 22: 289-338.

[12] Chen, J.S., Pan, C., Wu, C.T., and Liu, W.K. Reproducing kernel particle methods for large deformation analysis of non-linear structures. Comput. Methods Appl. Mech. Engrg. 1996; 139 (1-4): 195-227.

[13] Duarte CAM and Oden JT, An h-p adaptive method using clouds, Comput. Methods Appl. Mech. Engrg, 1996, 139, 237-262

[14] Liszka TJ, Duarte CAM, Tworzydlo WW, hp-meshless cloud method, Computer Methods in Applied Mechanics and Engineering, 1996, 139, 263-288

[15] Zhu T., Zhang J-D., and Atluri S.N. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Computational Mechanics. 1998; 21: 223-235.

[16] Atluri S.N., Sladeck J., Sladeck V., and Zhu T. The local boundary integral equation (LBIE) and its meshless implementation for linear elasticity. Computational Mechanics. 2000; 25: 180-198.

[17] Sladek, J., Sladeck, V., and Van Keer, R. Meshless local boundary integral equation for 2D elastodynamic problems. International Journal for Numerical Methods in Engineering. 2003: 235-249.

[18] Long, S. and Zhang. Q. Analysis of thin plates by the local boundary integral equation (LBIE) method. Engineering Analysis with Boundary Elements. 2002; 26: 707-718.

[19] Atluri, S.N. and Zhu, T. A new meshless Local Petrov-Galerkin approach in computational mechanics. Computational Mechanics. 1998; 22: 117-127.

[20] Atluri, S.N. and Zhu, T. New concepts in meshless methods. International Journal for Numerical Methods in Engineering. 2000; 47: 537-556.

[21] Atluri, S.N. and Zhu, T. The Meshless Local Petrov-Galerkin approach for solving problems in elasto-statics. Computational Mechanics. 2000; 225: 169-179.

[22] Liu, G.R. and Gu, Y.T. Meshless local Petrov-Galerkin (MLPG) method in combination with finite element and boundary element approaches. Computational Mechanics. 2000; 26(6):536-546.

[23] Mukherjee, Y.X. and Mukherjee, S. Boundary node method for potential problems. International Journal for Numerical Methods in Engineering. 1997; 40: 797-815.

[24] Chati, M.K. Mukherjee, S and Mukherjee, YX. The boundary node method for three-dimensional linear elasticity. International Journal for Numerical Methods in Engineering. 1999; 46: 1163-1184.

[25] Sukumar N, Moran B, Belytschko T, The natural element method, International Journal for Numerical Methods in Engineering, 1998, 43, 829-887

[26] Gu, Y.T. and Liu, G.R. A boundary point interpolation method for stress analysis of solids. Computational Mechanics. 2002; 28: 47-54.

[27] Liu, G.R. and Gu, Y.T. A local point interpolation method for 2D solid. International J. Struct. Eng. Mech. 2001;11(2):221-236

[28] Liu, G.R. and Gu, Y.T. A point interpolation method for 2D solid. International Journal for Numerical Methods in Engineering. 2001; 50: 937-951.

[29] Liu, G.R. and Yan, L. A study on numerical integration in element-free methods. Proceeding of 4th Asia-Pacific Conference on Computational Mechanics, Singapore.1999: 979-984.

[30] Sukumar N, Moran B, Black T, Belytschko T, An element-free Galerkin method for three-dimensional fracture mechanics, Computational Mechanics, 1997, 20, 170-175

[31] Lee, Sang-Ho and Yoon, Yong-Cheol. Numerical prediction of crack propagation by an enhanced element-free Galerkin method. Nuclear Engineering and Design.2004; 227:257-271.

[32] Xu Y, Saigal S, An element-free Galerkin analysis of steady dynamic growth of a mode I crack in elastic-plastic materials, International Journal of Solids and Structures, 1999, 36, 1045-1079

[33] Bobaru F, Mukherjee S, Shape sensitivity analysis and shape optimization using the element-free Galerkin method, Computer Methods in Applied Mechanics and Engineering, 2001, 190, 4319-4337

[34] Sladek V., Sladek J., Atluri S.N., and Van Keer R. Numerical integration of singularities in meshless implementation of local boundary integral equations. Computational Mechanics. 2000; 25: 394-403.

[35] Paris, F. and Canas J. *Boundary Element Method*. Oxford University Press, 1997.

[36] Stroud, A.H. and Secrest, D. *Gaussian Quadrature Formulas*. Prentice-Hall, 1966.

[37] Aliabadi MH, Hall WS, Weighted Gaussian methods for three-dimensional boundary element kernel integration, Comm. Appl. Numer. Methods, 1987, 3, 89-96

[38] Pina HLG, Fernandes, JLM, Brebbia CA, Some numerical integration formulae over triangles and squares with a 1/R singularity, Appl. Math. Modelling, 1981, 5, 209-211

[39] Kellogg, O.D. Foundations of Potential Theory, Dover Publications, New York, 1954, p.149

[40] Sarihan, V., and Mukherjee, S. Axisymmetric viscoplastic deformation by the boundary element method. Int. J. Solids and Structures. 1982; 18: 1113-1128.

[41] Lean MH and Wexler A, Accurate numerical integration of singular boundary element kernels over boundaries with curvature, Int. J. Numer. Method. Engng., 1985, 21, 211-228

[42] Nagarajan, A., and Mukherjee, S. A mapping method for numerical evaluation of 2D integrals with 1/r singularity. Computational Mechanics. 1993; 12: 19-26.

[43] Huang, Q., and Du, Q. An improved formulation for domain stress evaluation by boundary element method in elastoplastic problems. Proceedings of the China-US Seminar on Boundary Integral Equation and Boundary Finite Element Methods in Physics and Engineering, Xian, China. 1988.

[44] Cruse TA, Richardson JD, Non-singular Somigliana stress identities in elasticity, Int. J. Numer. Methods Engng., 1996, 39, 3273-3304

[45] Rudolphi, T.J. The use of simple solutions in the regularization of hypersingular boundary integral equations, Mathematical and Computer Modeling. 1991; 15: 269-278

[46] Lachat JC, Watson JO, Effective numerical treatment of boundary integral equations: A formulation for three-dimensional elastostatics, Int. J. Numer. Methods Engng., 1976, 10, 991-1005

[47] Guiggiani M, Gigante A, A general algorithm for multidimensional Cauchy principal value integrals in the boundary element method, ASME J. Appl. Mech., 1990, 57, 906-915

[48] Guiggiani M, The evaluation of Cauchy principal value integrals in the boundary element method - a review, Math. Comput. Modeling, Special Issue on BIEM/BEM, 1991, 15, 3-5, 175-184

[49] Mukherjee, Y.X. and Mukherjee, S. On boundary conditions in the element-free Galerkin method, Computational Mechanics. 1997; 19: 264-270

[50] Zhu, T. and Atluri, S.N. A modified collocation method and a penalty foumulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput. Mech. 1998; 21(3): 211-222.

[51] Wagner, GJ and Liu WK, Application of essential boundary conditions in mesh-free methods: a corrected collocation method, International Journal for Numerical Methods in Engineering, 2000, 47, 1367-1379

[52] Wu CC and Plesha ME, Essential boundary condition enforcement in meshless methods: boundary flux collocation method, Int. J. Numer. Meth. Engng, 2002, 53, 499-514

[53] Herault, C. and Marechal, Y. Boundary and interface conditions in meshless methods. *IEEE Transactions on Magnetics.* 1999; 35(3): 1450-1453.

[54] Fernandez-Mendez, Sonia and Huerta Antonio. Imposing essential boundary conditions in mesh-free methods. Comput. Methods Appl. Mech. Engrg. 2004;193: 1257-1275.

[55] Gavete, L., Benito, J.J., Falcon S. and Ruiz A. Implementation of essential boundary conditions in a meshless method. *Commun. Numer. Meth. Engng.* 2000; 16: 409-421.

[56] J. Nitsche, Uber ein variations zur losung von dirichlet-problemen bei verwendung von teilraumen die keinen randbedingungen unterworfen sind, Adh. Math. Se. Univ. 1970, 36, 9-15

[57] Arnold DN, Brezzi F, Cockburn B, Marini LD, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIMA J. Numer. Anal., 2002, 39, 1749-1779

[58] Belytschko, T., Organ, D., and Krongauz, Y. A coupled finite element-element-free Galerkin method. Computational mechanics. 1995; 17: 186-195.

[59] Rao, B.N. and Rahman, S. A coupled meshless-finite element method for fracture analysis of cracks. Pressure Vessels and Piping. 2001; 78: 647-657.

[60] Lou, L.L. and Zeng, P. FE-Meshless Coupling Method for 2D Crack Propagation. Key Engineering Materials. 2003; 233-236: 169-174.

[61] Hegen D., Element-free Galerkin methods in combination with finite element approaches, Computer Methods in Applied Mechanics and Engineering, 1996, 19, 120-135

[62] Gosz, J. and Liu, W.K. Admissible approximations for essential boundary conditions in the reproducing kernel particle method. Comput. Mech. 1996; 19(2):120-135.

[63] Gunther, F. C. and Liu, W. K. Implementation of boundary conditions for meshless methods. Computer Methods in Applied Mechanics and Engineering 1998; 163: 205-230.

[64] Zhang X, Liu X, Lu MW and Chen Y, Imposition of essential boundary conditions by displacement constraint equations in meshless methods, Communications in Numerical Methods in Engineering, 2001, 17, 165-178

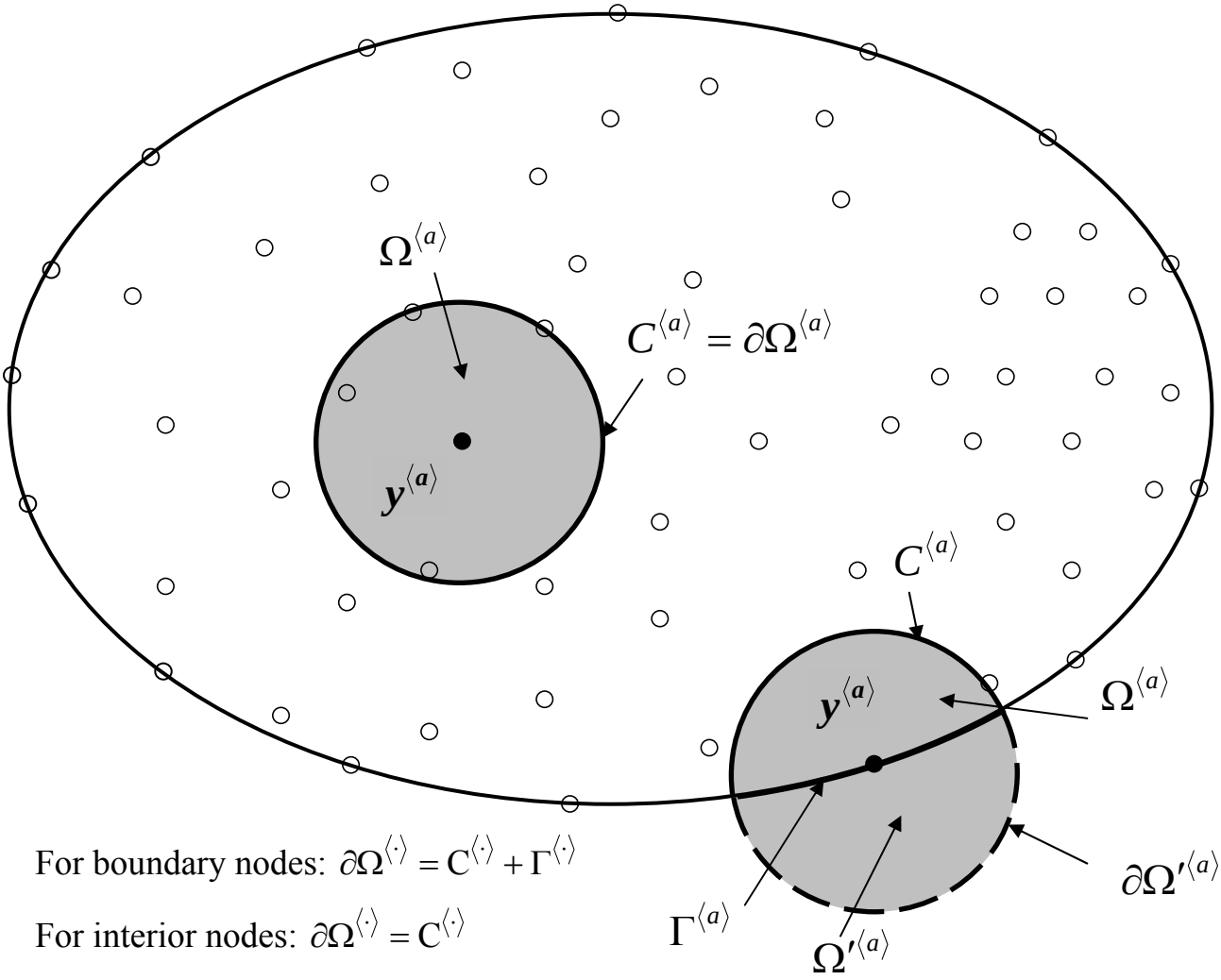
[65] Onate E, Idelsohn S, Zienkiewicz OC, Taylor RL, A finite point method in computational mechanics. Applications to convective transport and fluid flow. International Journal for Numerical Methods in Engineering, 1990, 39, 3839-3866

[66] Taylor, R.L., Beresford, P.J., and Wilson, E.L. A non-conforming element for stress analysis. Int. J. Numer. Anal. Meth. Engng. 1976; 10: 1211-1219.

[67] Timoshenko, S.P. and Goodier, J.N. *Theory of elasticity*. McGraw-Hill Book Compny, 1970.

[68] ANSYS 5.6 Documentation, ANSYS Incorporated, 1999.

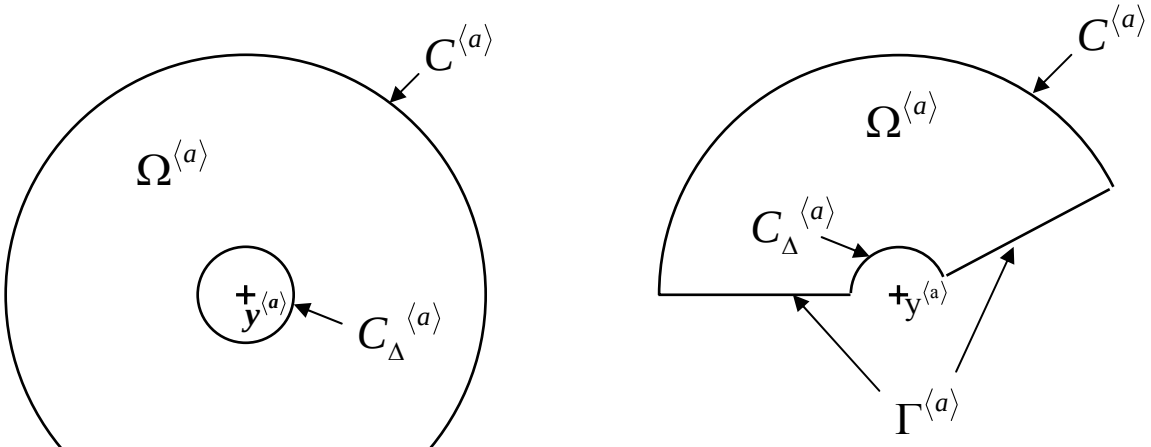
Figure 2.1 Schematic diagram showing the sub-domain for an interior or a boundary node $\mathbf{y}^{\langle a \rangle}$



For boundary nodes: $\partial\Omega^{\langle \cdot \rangle} = \mathrm{C}^{\langle \cdot \rangle} + \Gamma^{\langle \cdot \rangle}$

For interior nodes: $\partial\Omega^{\langle \cdot \rangle} = \mathrm{C}^{\langle \cdot \rangle}$

Figure 2.2 Exclusion of a tiny sphere Ω_Δ of radius Δ centered a node for removing the strong singularity



When $\mathbf{y}^{<a>}$ is an interior node

$$\partial\Omega^{\langle a \rangle} = C^{\langle a \rangle} + C_\Delta{}^{\langle a \rangle} + \Gamma^{\langle a \rangle}$$

with $\Gamma^{\langle a \rangle} = \varnothing$

When $\mathbf{y}^{<a>}$ is a boundary node

$$\partial\Omega^{\langle a \rangle} = C^{\langle a \rangle} + C_\Delta{}^{\langle a \rangle} + \Gamma^{\langle a \rangle}$$

Figure 2.3 Schematic diagram showing the internal boundary angle $\theta = \theta_2 - \theta_1$ at node $\mathbf{y}^{\langle a \rangle}$ on the boundary

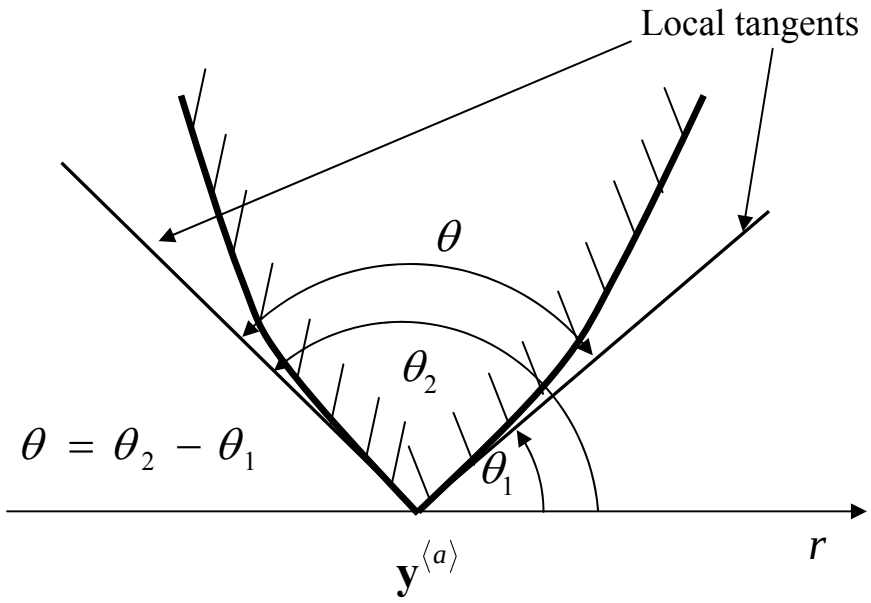


Figure 2.4 Schematic diagram illustrating the local sub-domains and support domains for node $\mathbf{y}^{\langle a \rangle}$ and node $\mathbf{y}^{<b>}$

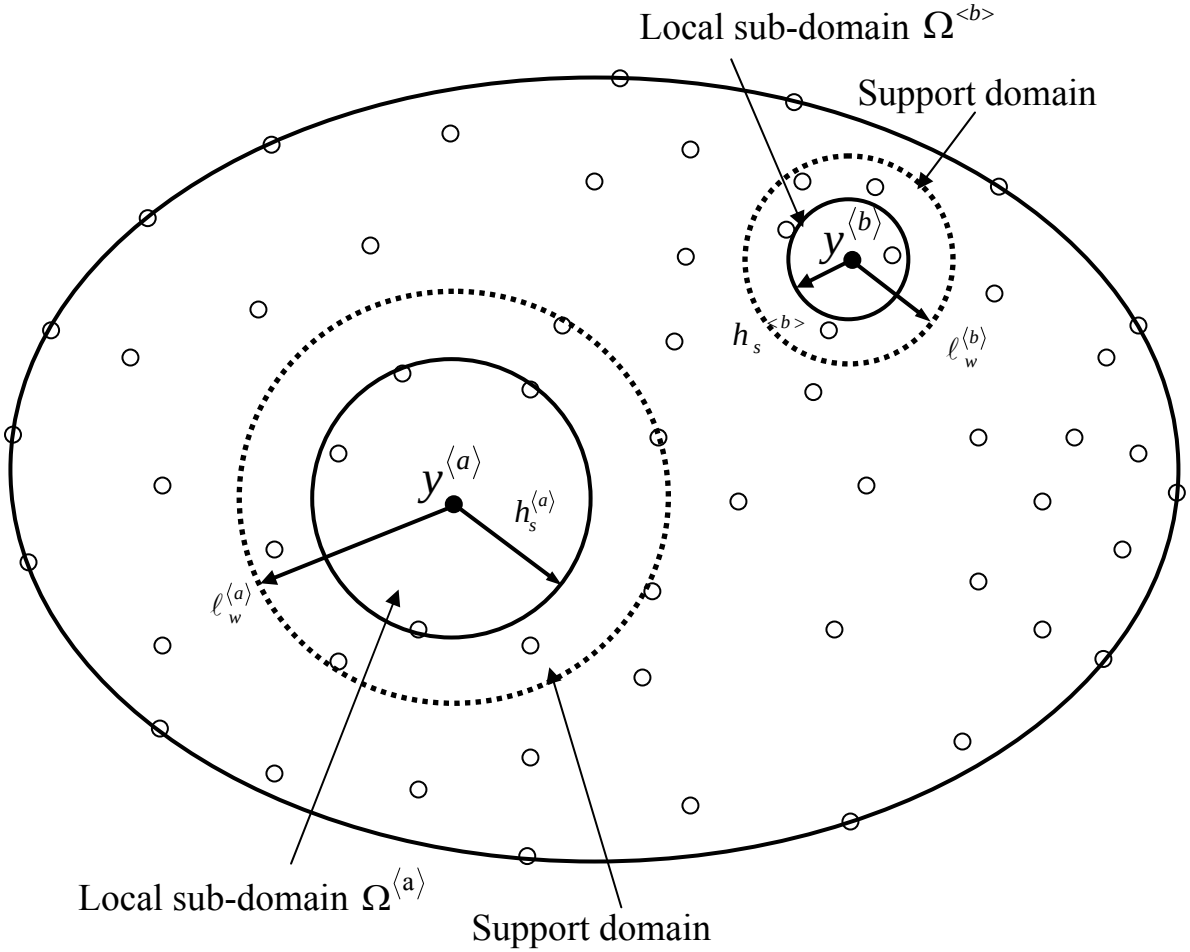


Figure 2.5 Schematic diagram showing the domain of influence for node $\mathbf{y}^{\langle a \rangle}$ and the domain of definition for a point $\mathbf{x}$

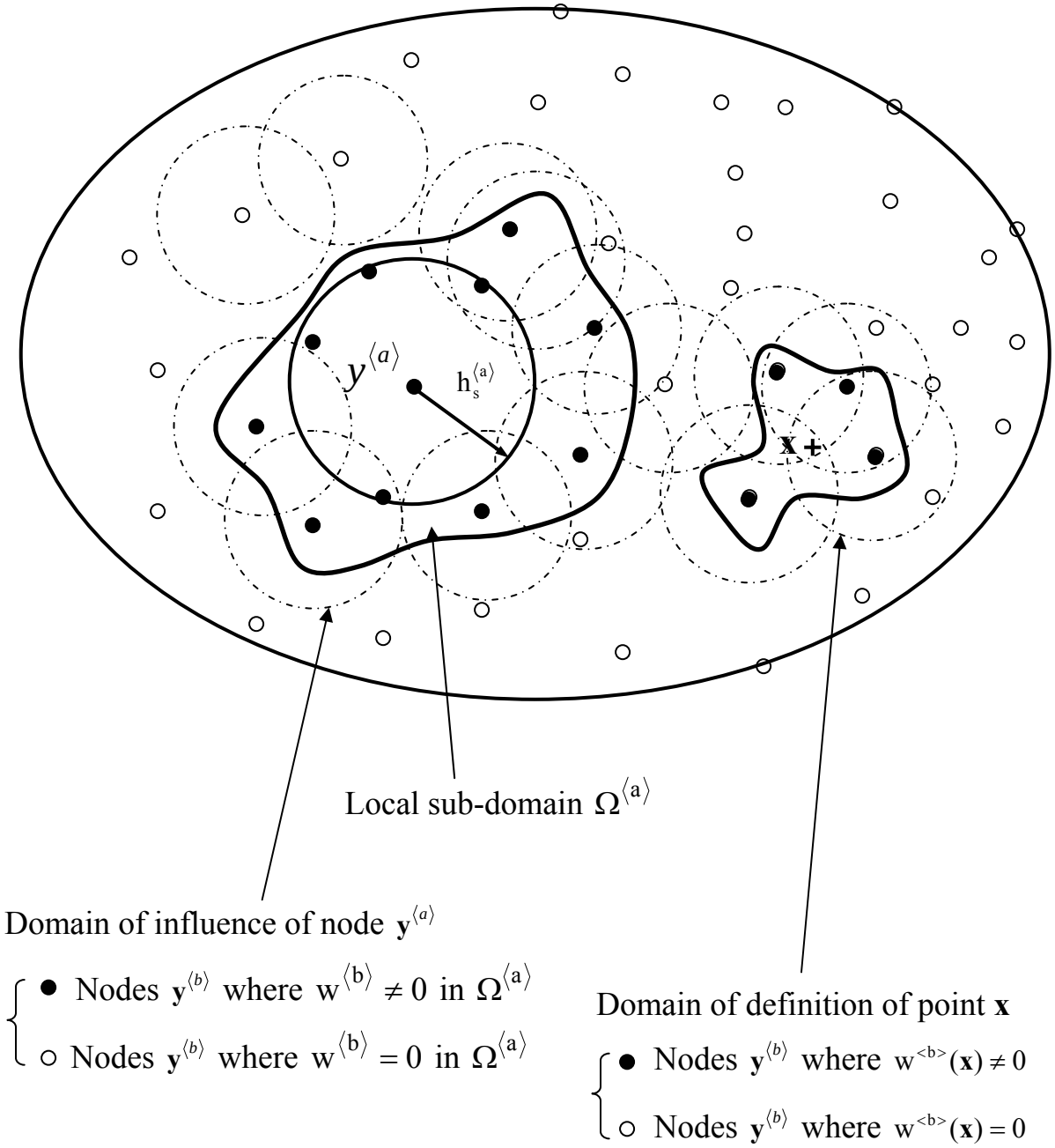
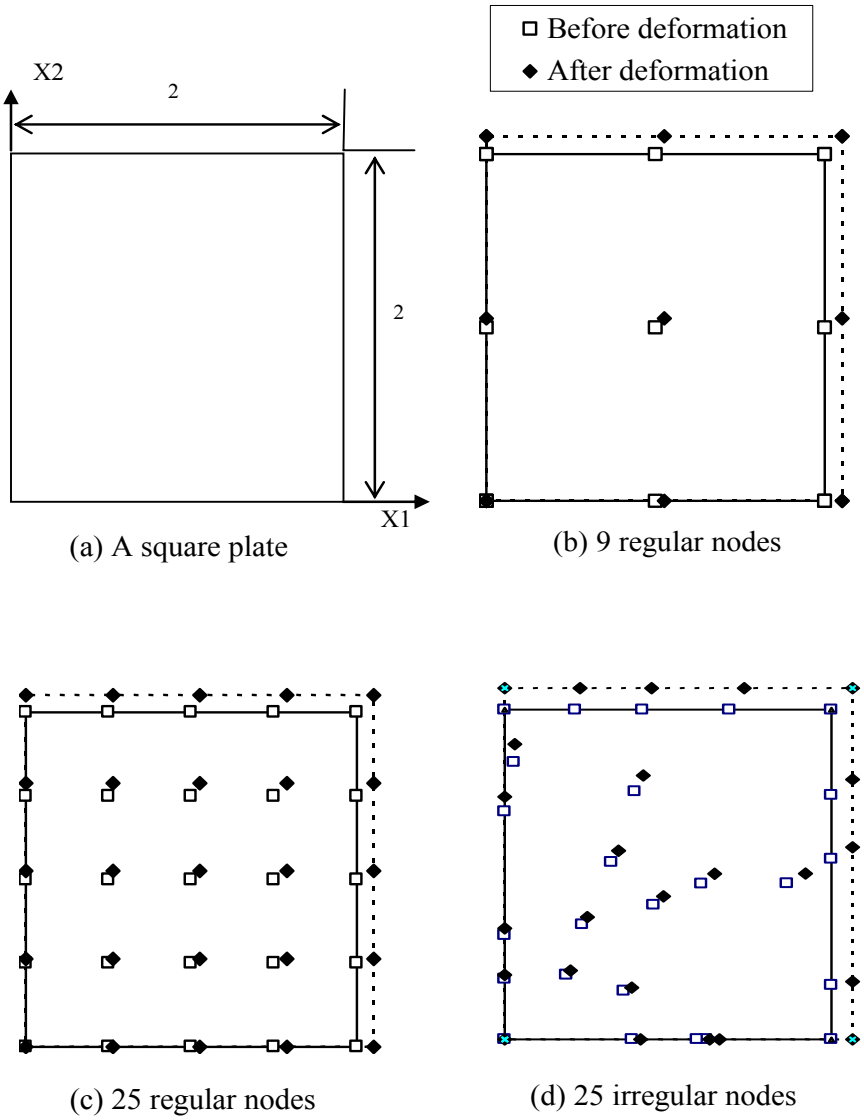


Figure 2.6 (a) A square plate for the patch test. (b) Meshless model with 9 regular nodes. (c) Meshless model with 25 regular nodes. (d) Meshless model with 25 irregular nodes



(a) A square plate

(b) 9 regular nodes

(c) 25 regular nodes

(d) 25 irregular nodes

Figure 2.7 High order patch test-- patch with 28 uniformly distributed nodes

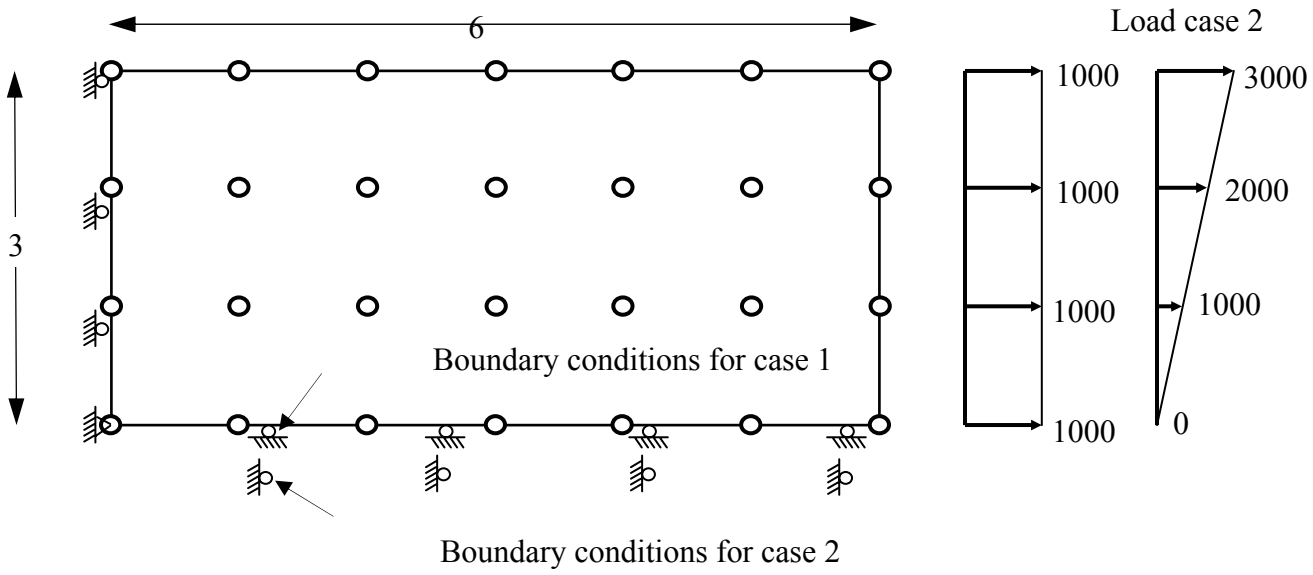


Figure 2.8 High order patch test-- Patch with 14 non-uniformly distributed nodes

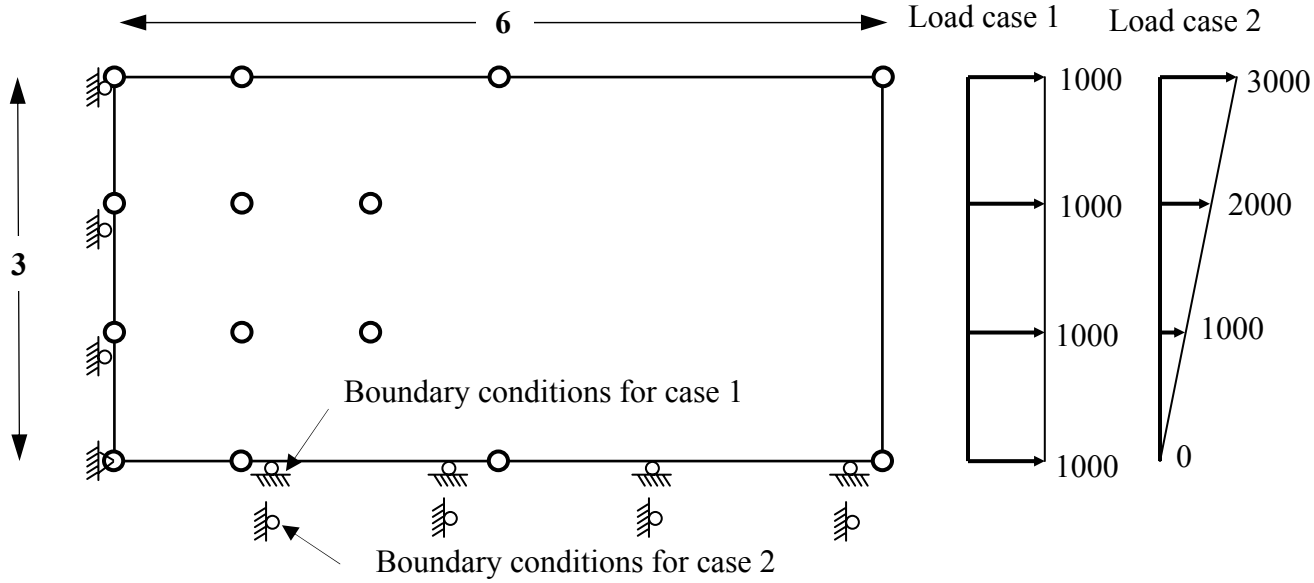
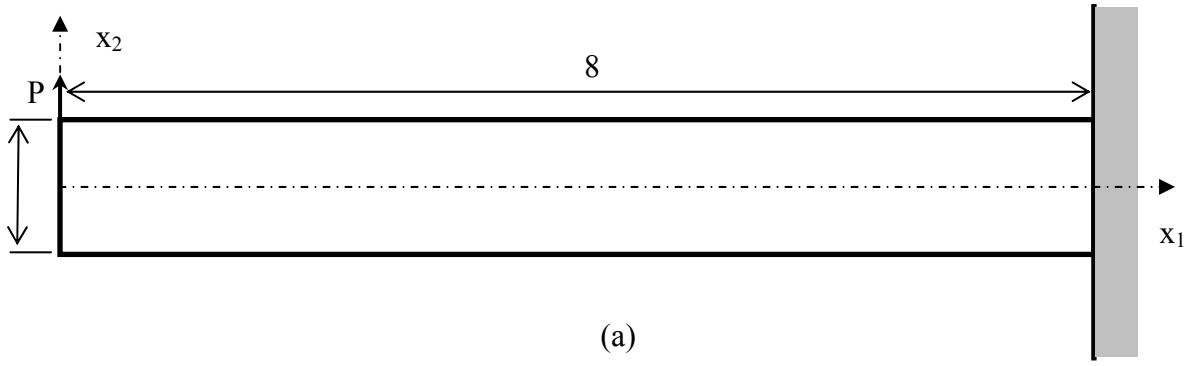
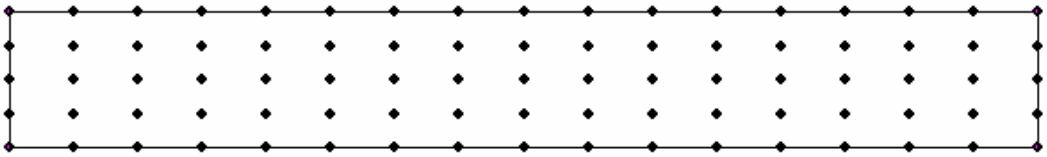


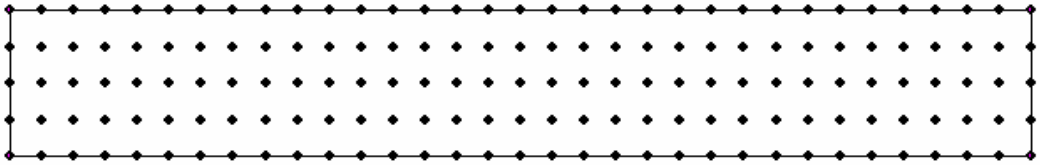
Figure 2.9 A cantilever beam subjected to end load and three meshless models



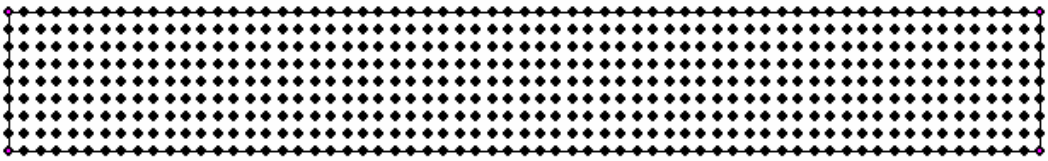
(a)



(b)



(c)



(d)

Figure 2.10 Deformed meshless model for the cantilever beam with 85 nodes

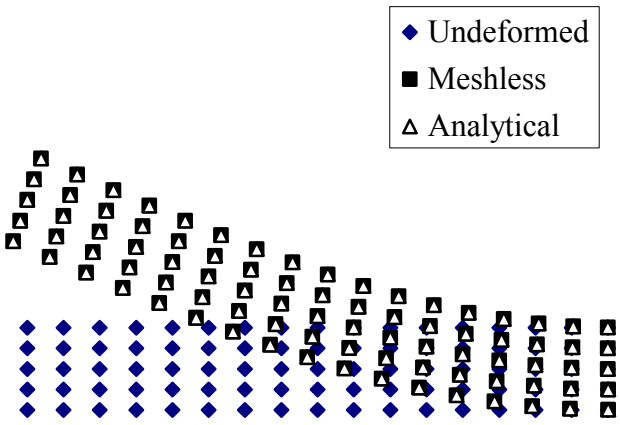


Figure 2.11 The effects of monomial basis (linear, quadratic, and cubic) for the 85-node meshless model using three weight functions

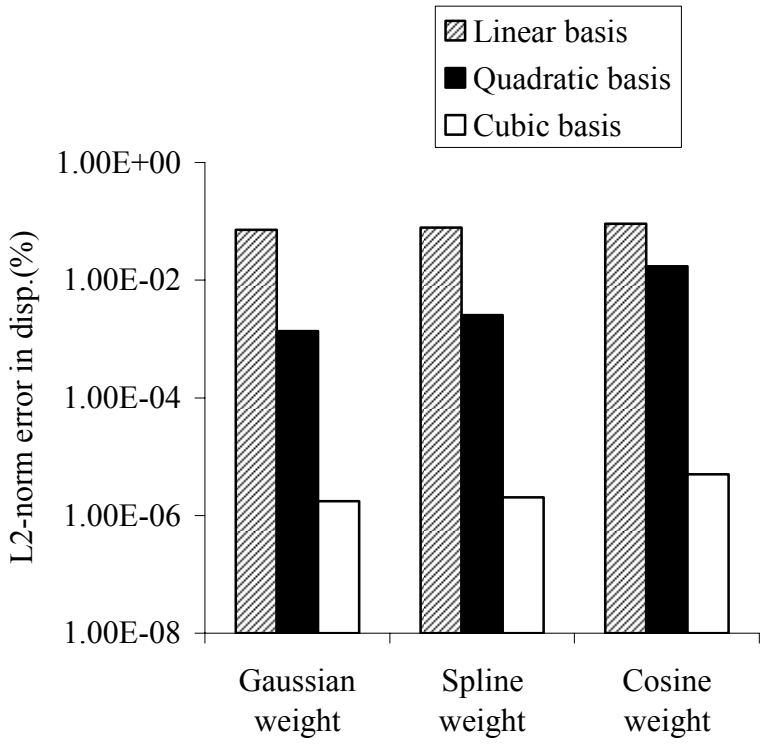


Figure 2.12 Comparison of percentage error of the displacement at the point (0, 0.5) between the original LBIE [16] and our current method

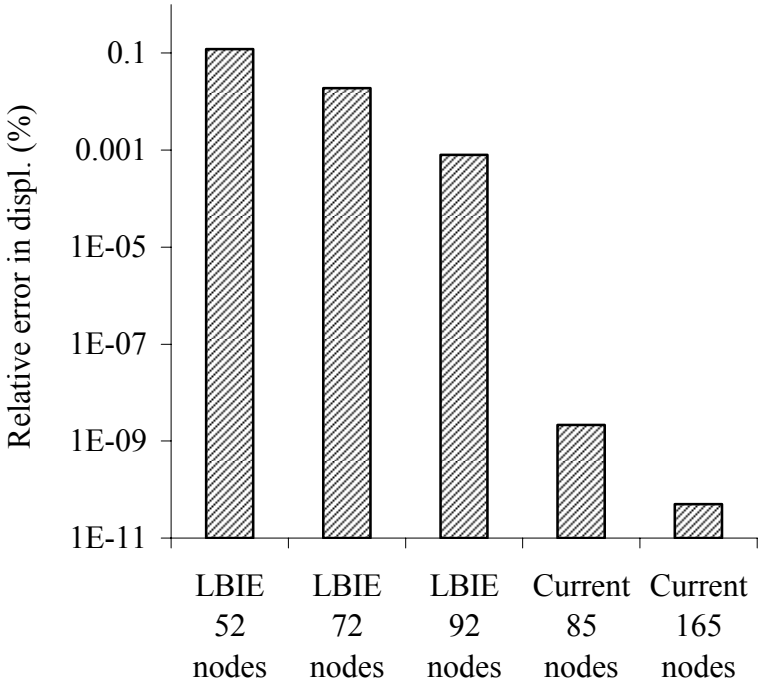


Figure 2.13 Comparison of the percentage error of the shear stress at the point (4, 0)

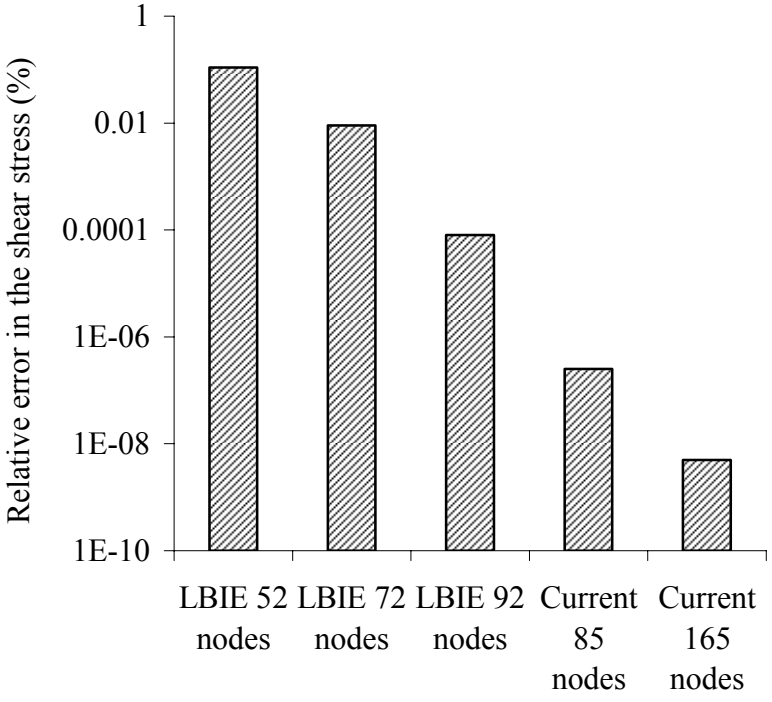


Figure 2.14 The effects of the monomial basis on the shear stress along the vertical cross-section at $x_1 = 4$

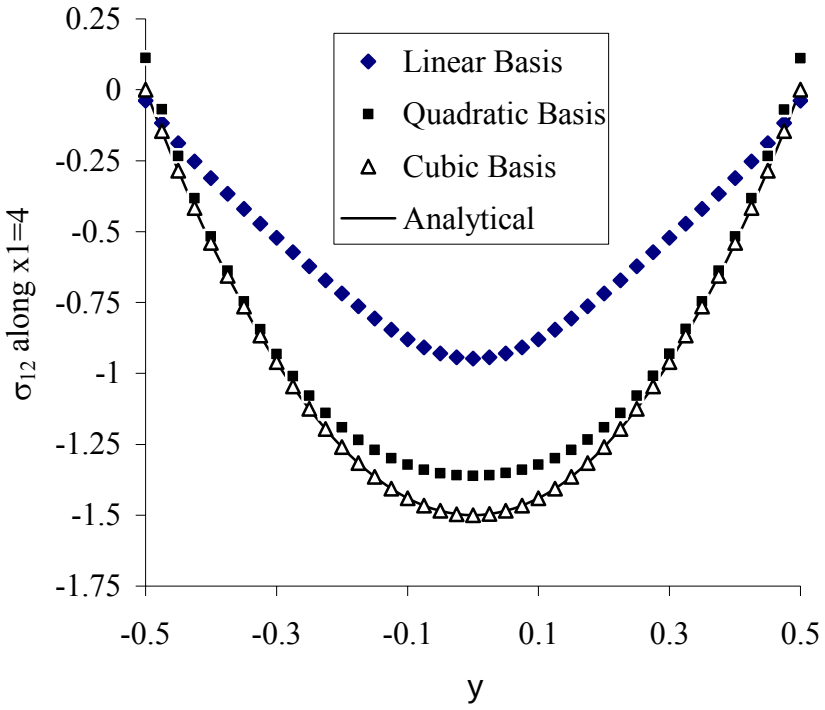


Figure 2.15 Comparison of the shear stress between the meshless method and FEM

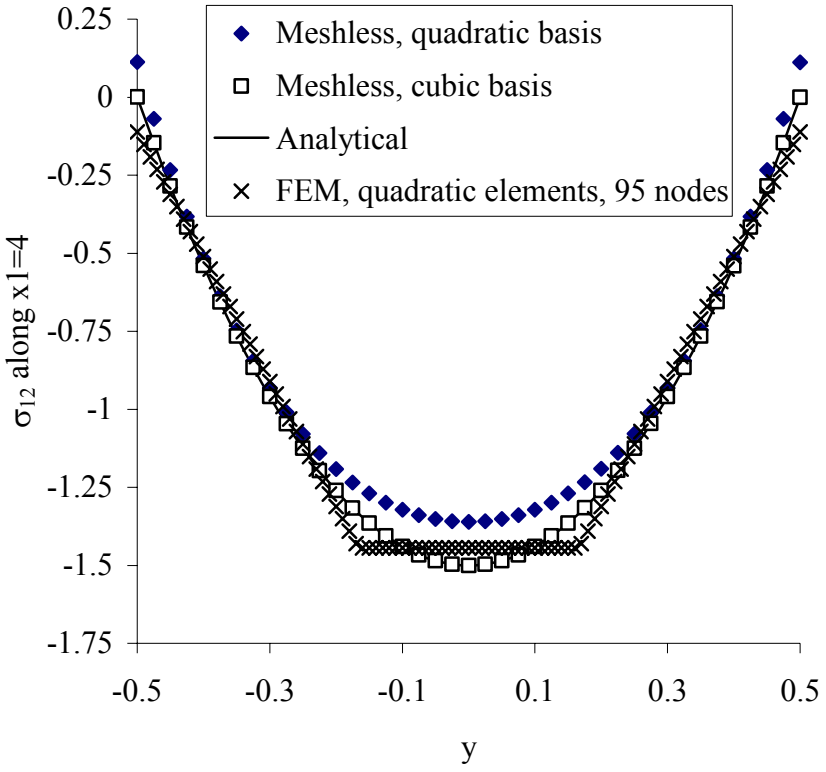
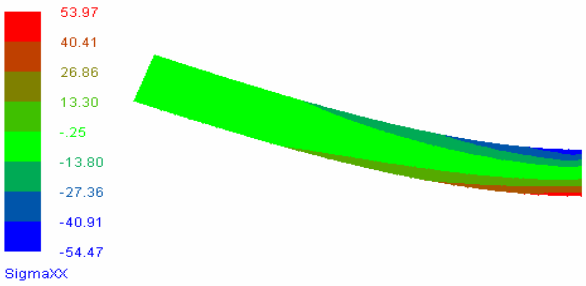
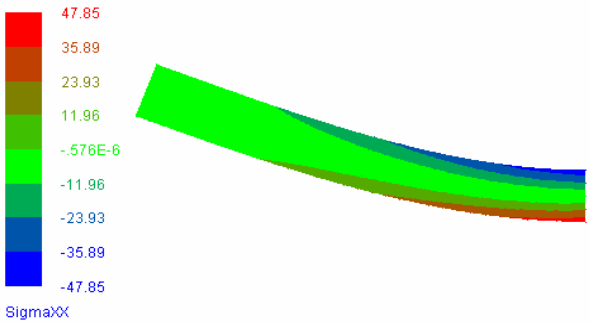


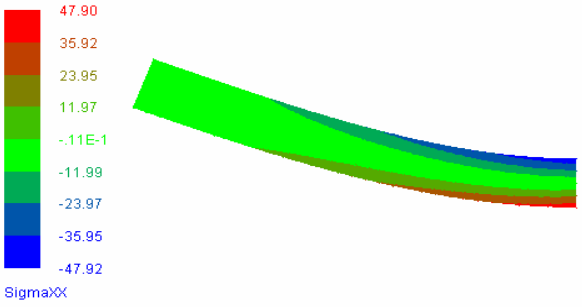
Figure 2.16 Contour plots of σ_{11} for meshless model (165 nodes) using spline weight function (a) Linear basis is used. (b) Quadratic basis is used. (c) Cubic basis is used



(a)

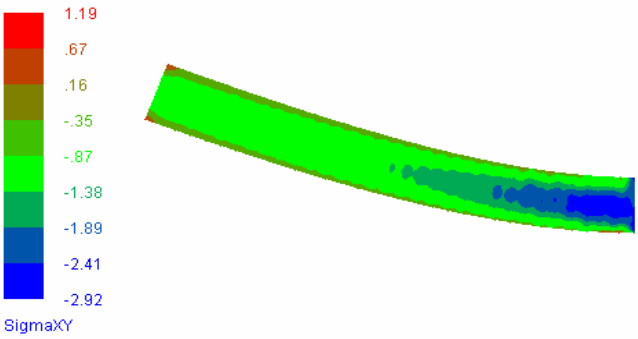


(b)

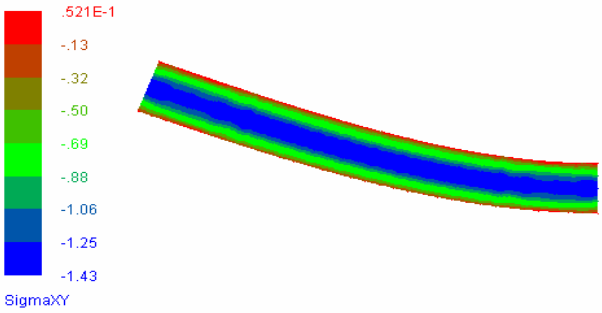


(c)

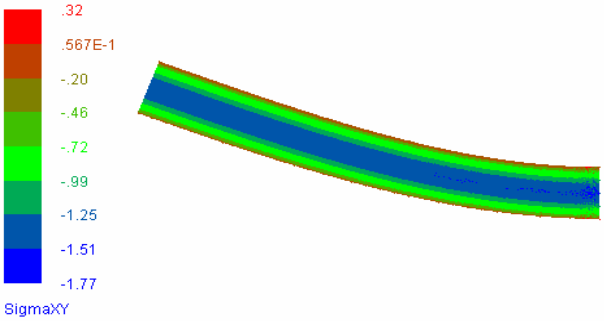
Figure 2.17 Contour plots of σ_{12} for meshless model (165 nodes) using spline weight function (a) Linear basis is used. (b) Quadratic basis is used. (c) Cubic basis is used.



(a)

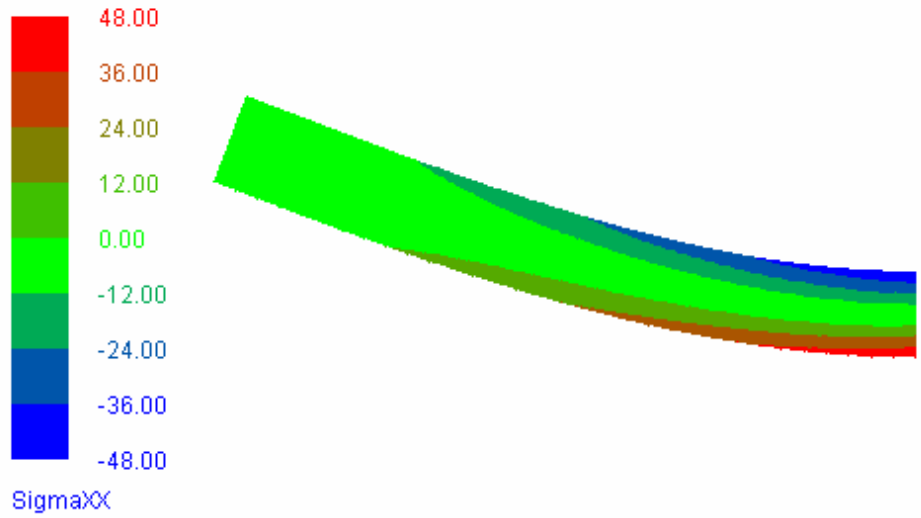


(b)

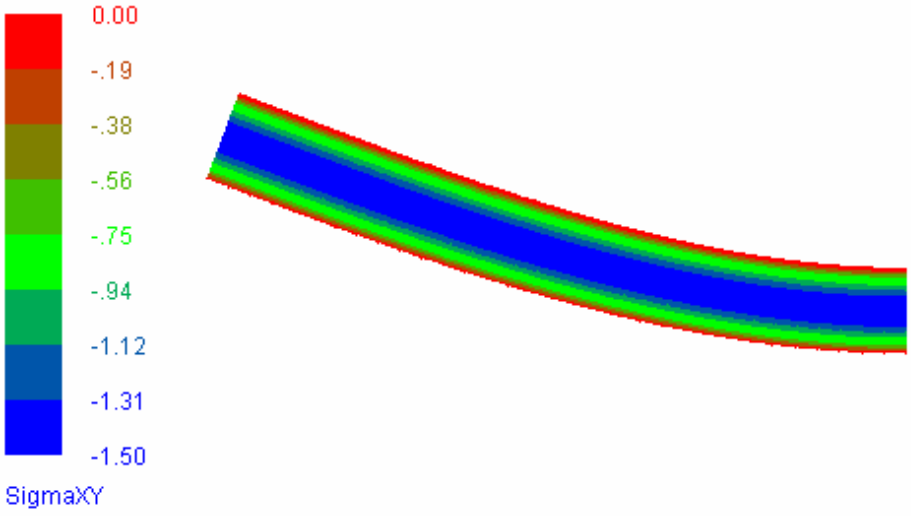


(c)

Figure 2.18 Contour plots of σ_{11} and σ_{12} from analytical stress. (a) σ_{11} contour plot. (b) σ_{12} contour plot



(a)



(b)

Figure 2.19 The upper right quadrant of the plate with a hole. Symmetric boundary conditions are applied to the left and bottom edges

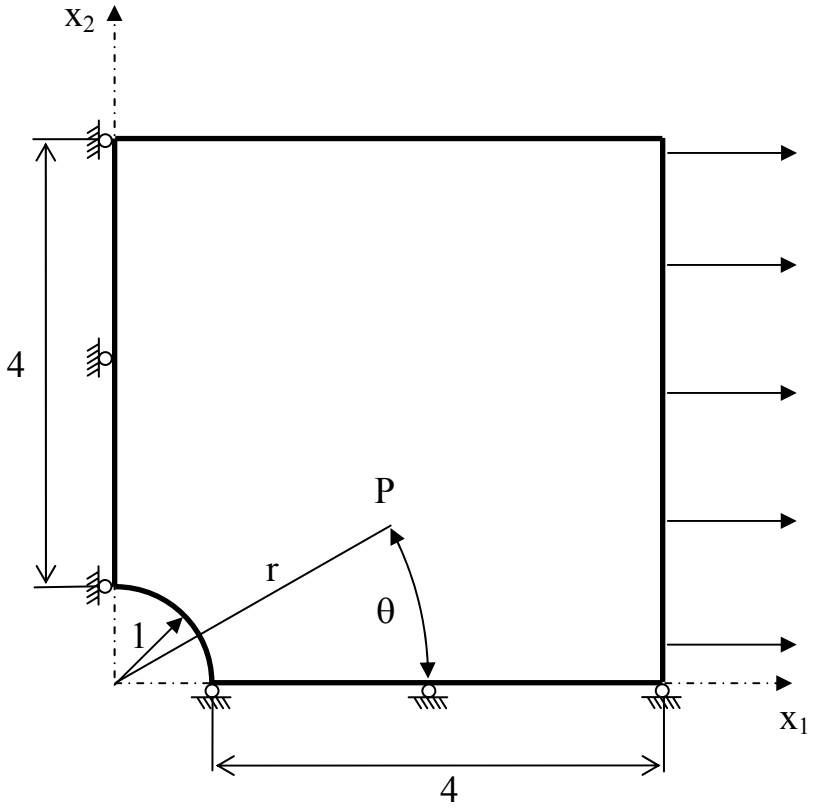


Figure 2.20 Deformed meshless model (empty squares) for case 1. Analytical results (solid triangles) are plotted for comparison

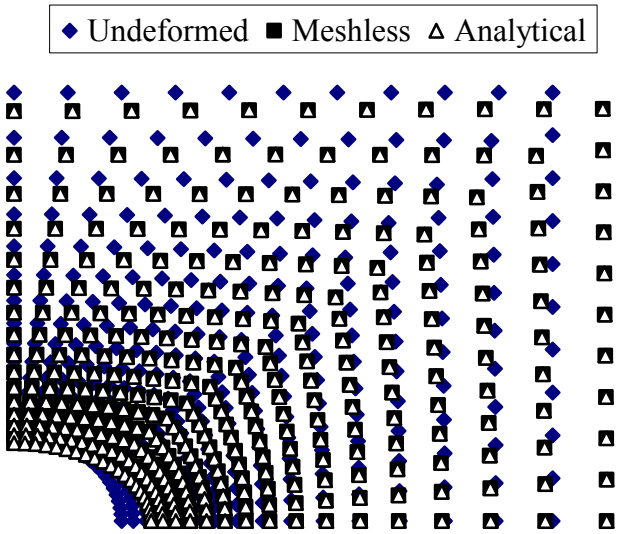


Figure 2.21 Deformed meshless model (empty squares) for case 2. Analytical results (solid triangles) are plotted for comparison

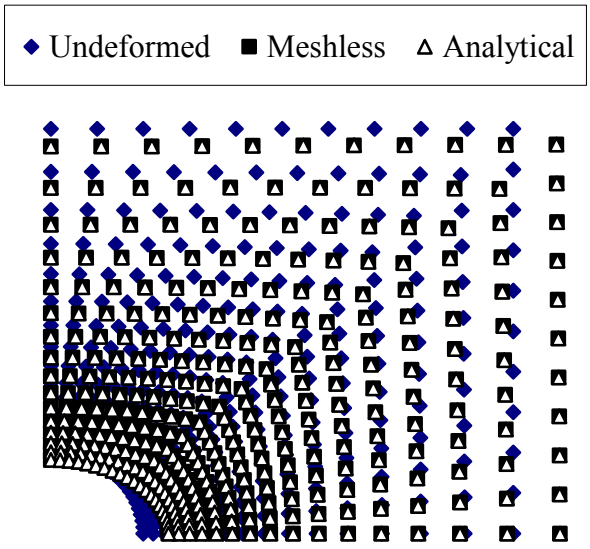


Figure 2.22 The distribution of normal stress σ_{11} along $x_1 = 0$ of the 336-node model for load case 1

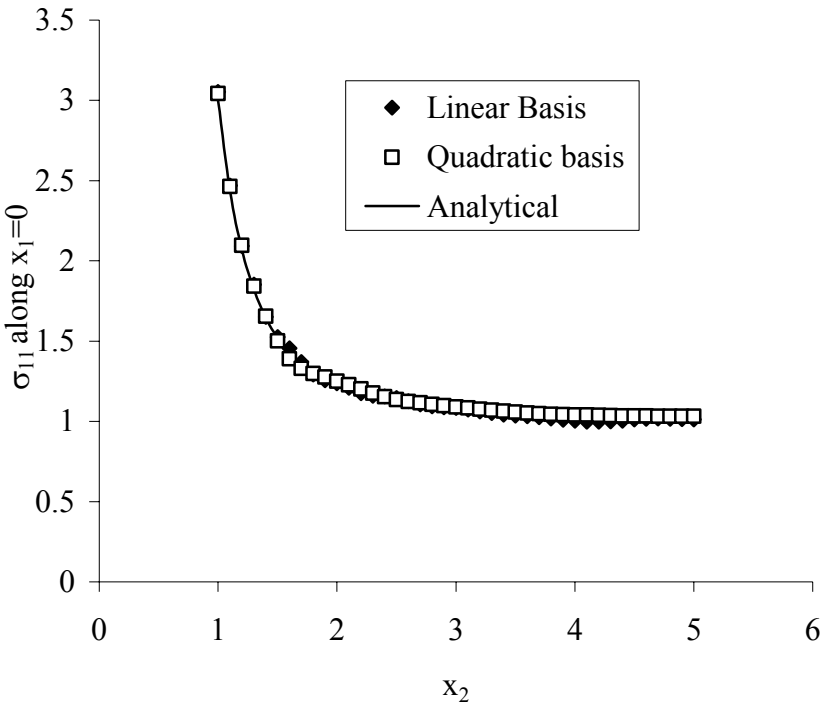


Figure 2.23 The distribution of normal stress σ_{11} along $x_1 = 0$ of the 336-node model for load case 2

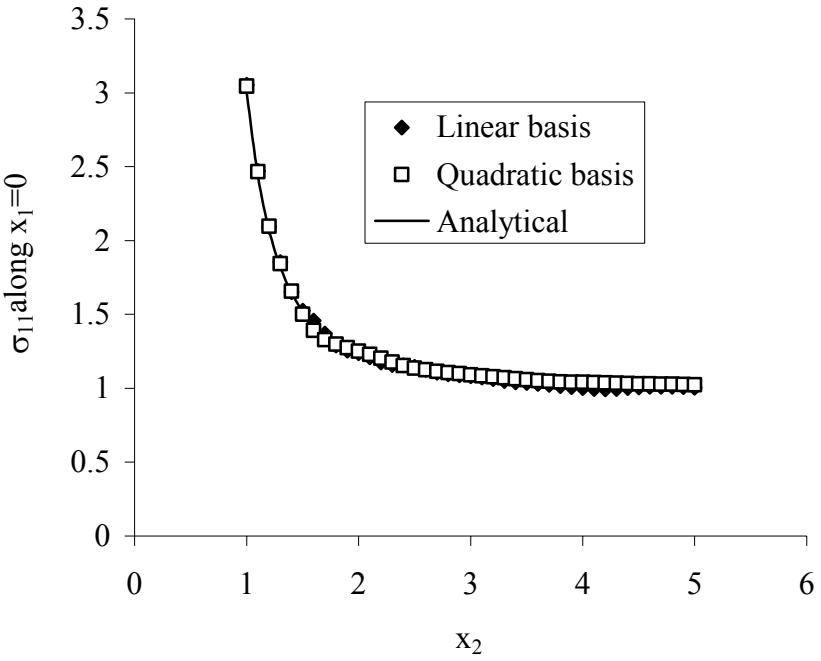


Figure 2.24 The distribution of normal stress σ_{11} along $x_1 = 0$ of the 1271-node model for load case 2

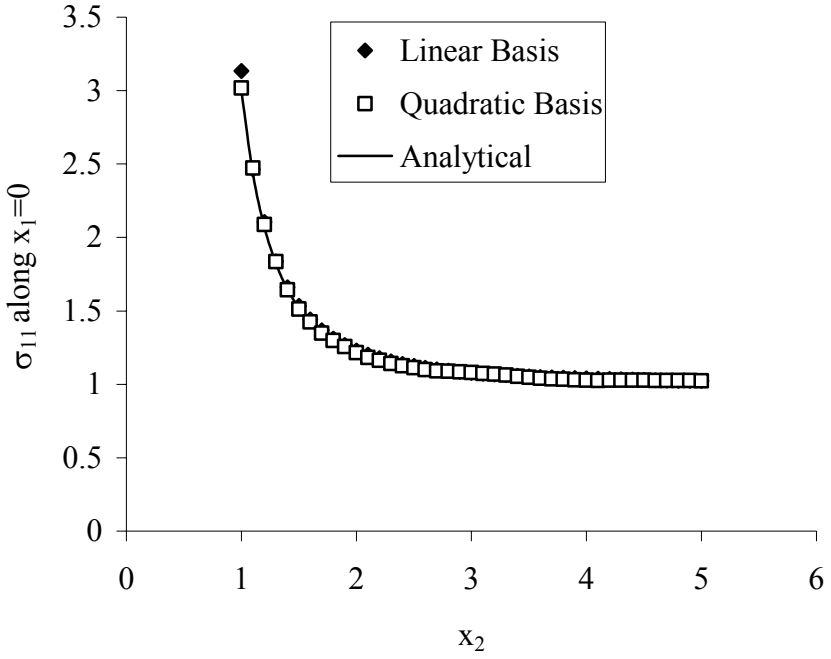


Figure 2.25 The distribution of normal stress σ_{11} along $x_1 = 0$ from the original LBIE [16] with 120 nodes

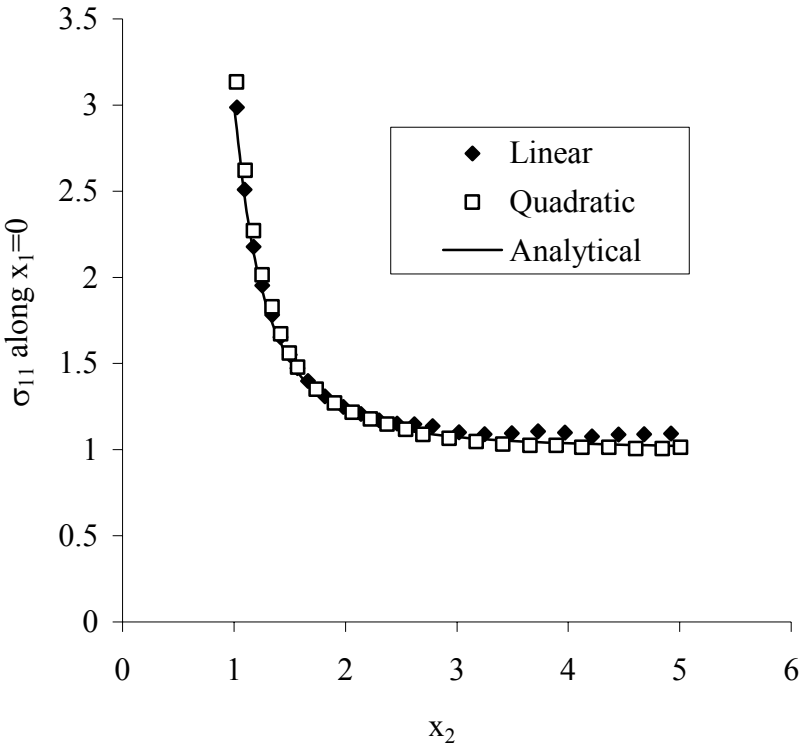


Figure 2.26 Plots of σ_{11} for meshless model (336 nodes) with spline weight function with linear basis used

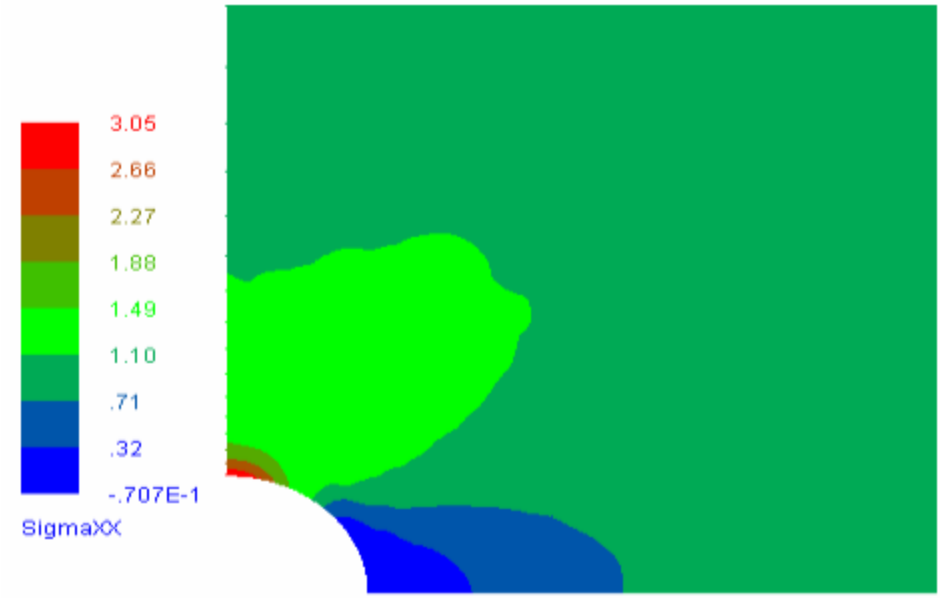


Figure 2.27 Plots of σ_{11} for meshless model (336 nodes) with spline weight function with quadratic basis used

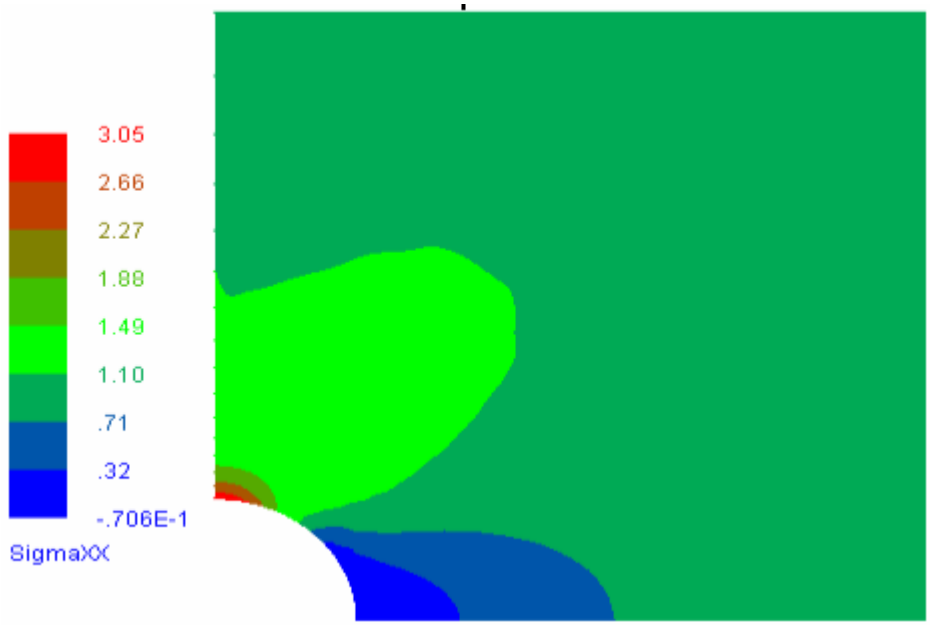


Figure 2.28 Contour plots of σ_{11} for FEM model with linear basis used (336 nodes)

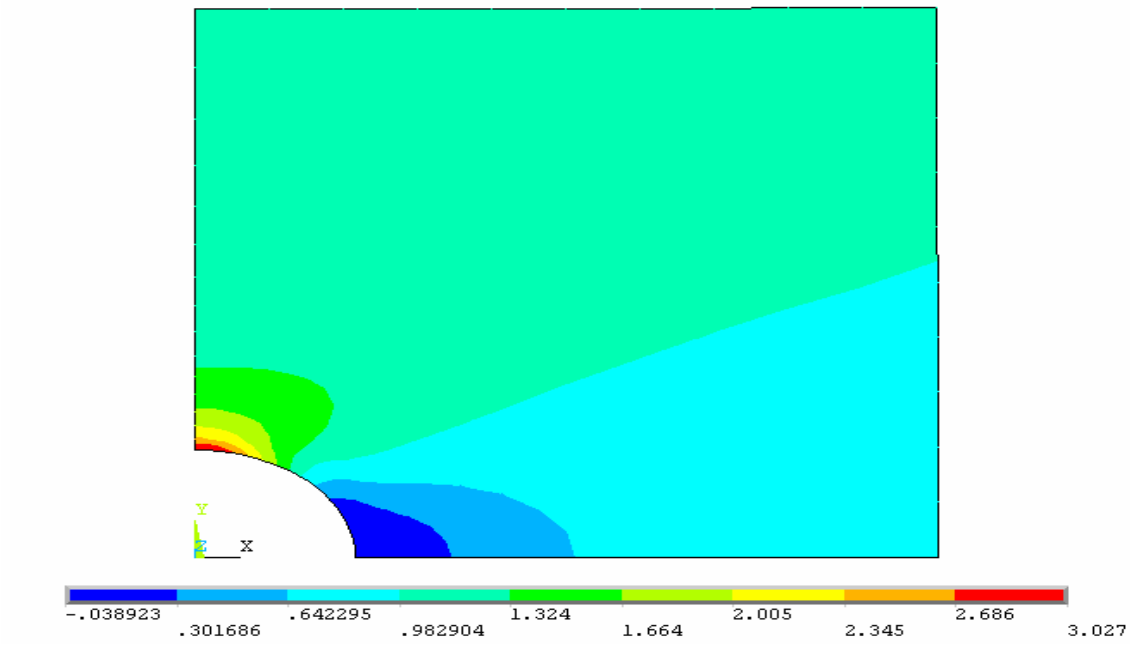


Figure 2.29 Contour plots of σ_{11} for FEM model with quadratic basis used (336 nodes)

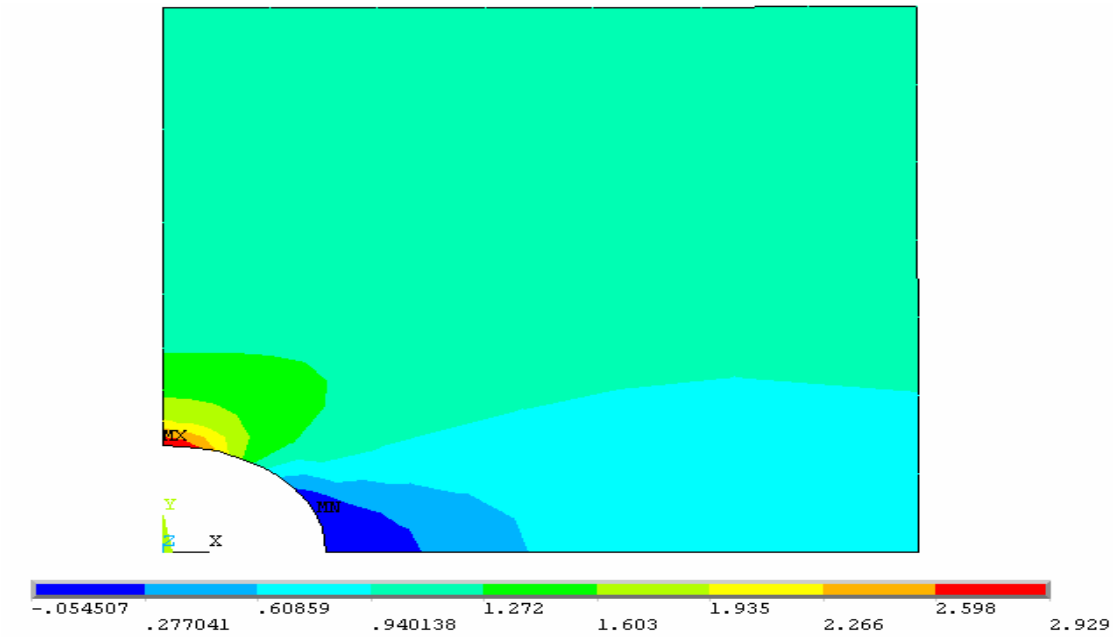


Figure 2.30 Contour plots of σ_{12} for meshless model (336 nodes) with linear basis used

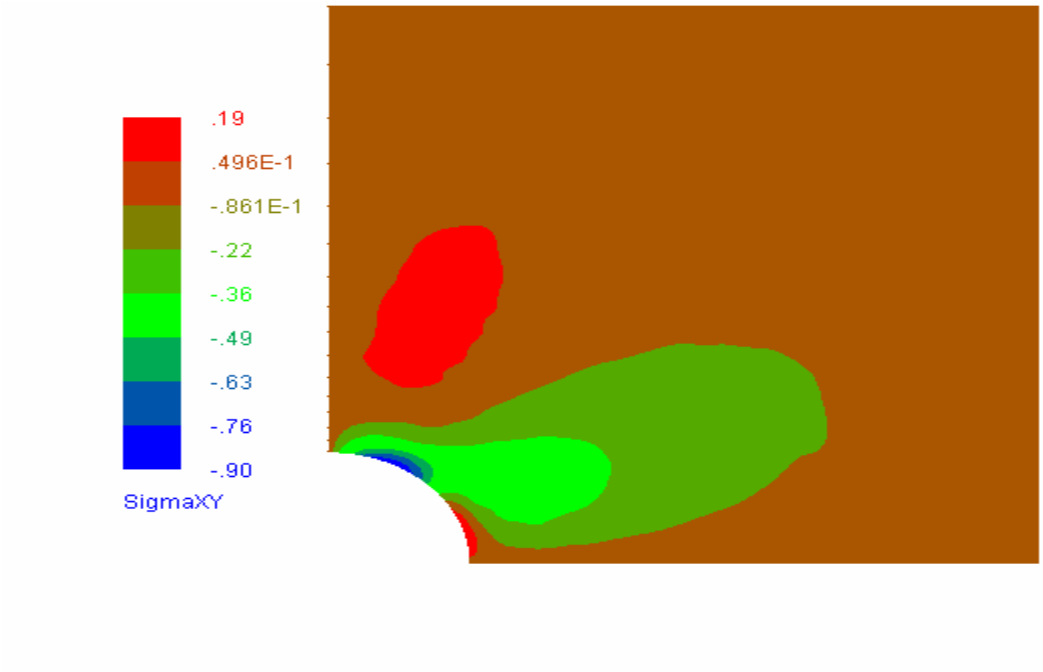


Figure 2.31 Contour plots of σ_{12} for meshless model (336 nodes) with quadratic basis used

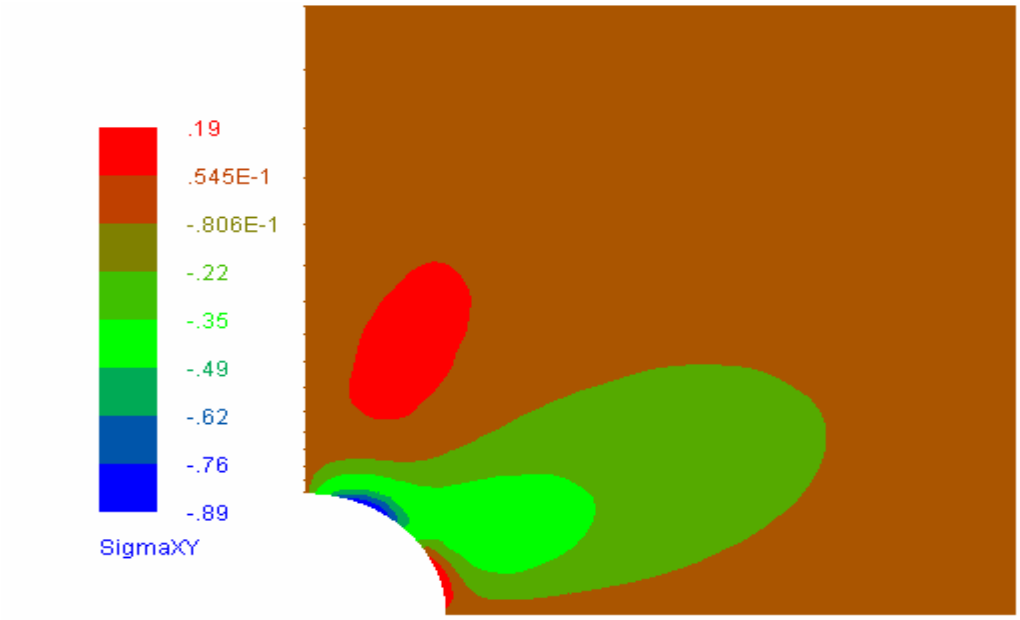


Figure 2.32 Contour plots of σ_{12} for FEM model with linear basis used (336 nodes)

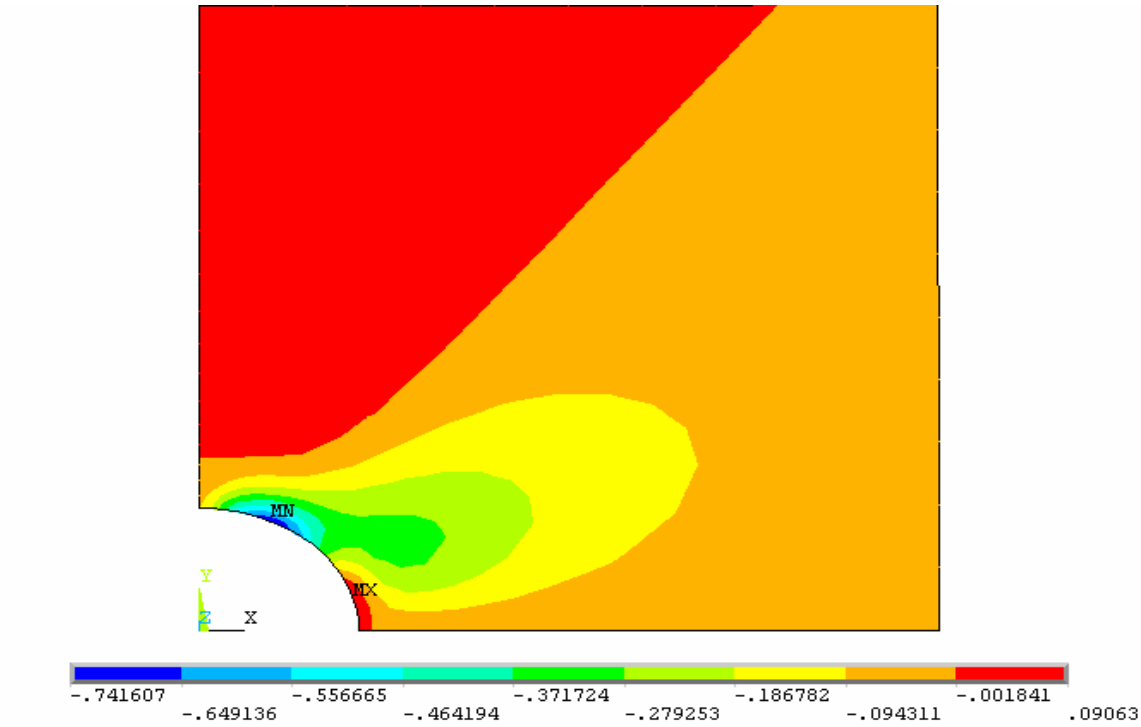


Figure 2.33 Contour plots of σ_{12} for FEM model with quadratic basis used (336 nodes)

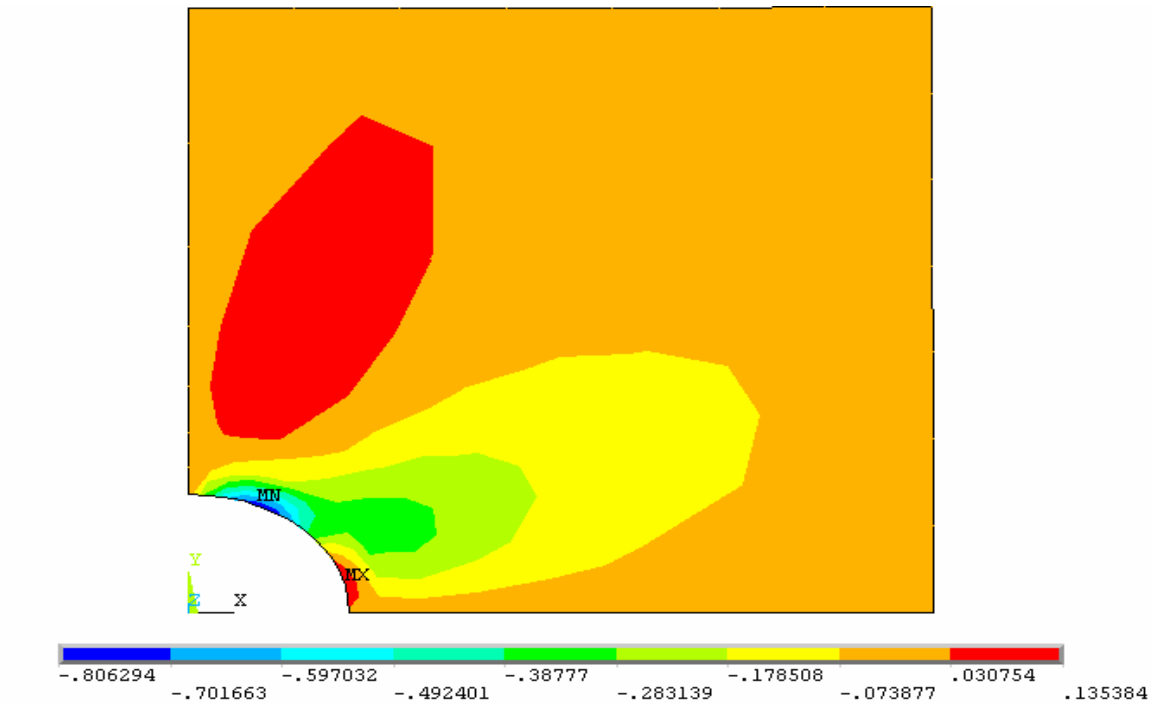


Figure 2.34 σ_{11} contour plot for analytical solutions

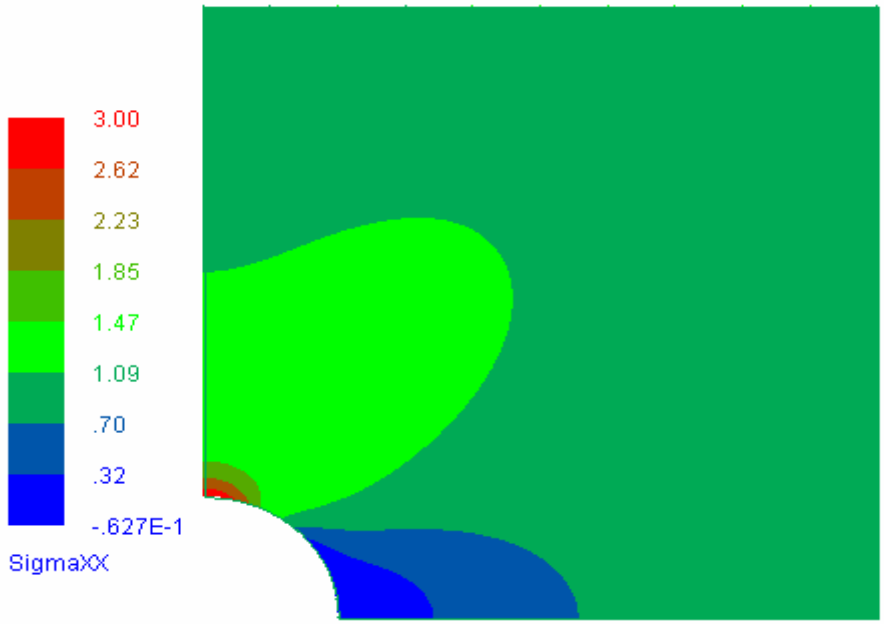


Figure 2.35 σ_{12} contour plot for analytical solutions

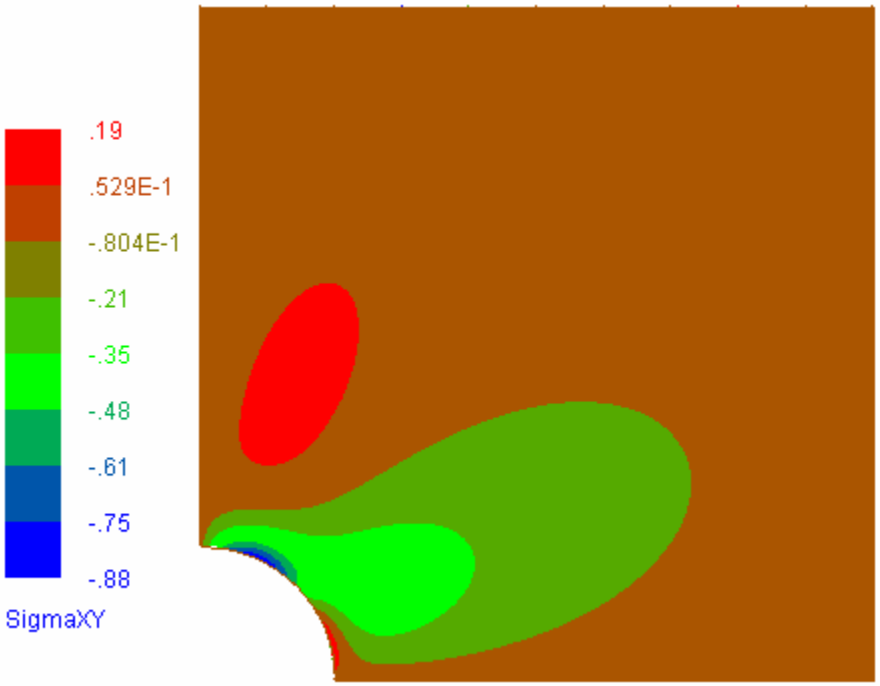


Table 2.1 Special numerical integration for functions containing logarithmic singularity (8 integration points)

$$\int_0^1 f(r)\ln\left(\frac{1}{r}\right)dr = \sum_{i=1}^{N} w_i f(r_i)$$

Abscissas (r_i)	Weights (w_i)
0.0133 2024 4160 8925	0.1644 1660 4728 0030
0.0797 5042 9013 8949	0.2375 2561 0023 3060
0.1978 7102 9326 1880	0.2268 4198 4431 9190
0.3541 5399 4351 9090	0.1757 5407 9006 0700
0.5294 5857 5234 9170	0.1129 2403 0246 7590
0.7018 1452 9939 1000	0.0578 7221 0717 7821
0.8493 7932 0441 1070	0.0209 7907 3742 1330
0.9533 2645 0056 3600	0.0036 8640 7104 0276

Table 2.2 L_2-norm errors for patch test

	9 nodes	25 nodes-regular	25 nodes-irregular
Case 1	2.1E-16	6.4E-16	9.3E-15
Case 2	6.7E-11	1.6E-10	3.6E-10

Table 2.3 L_2-norm errors for higher order patch test

	14 nodes	28 nodes
Case 1	1.4E-08	2.1E-8
Case 2	1.0E-7	1.1E-7

Table 2.4 L_2 **-norm errors for bending of a cantilever beam**

No. of nodes	85	165	585
Linear	0.078	0.018	0.020
Quadratic	0.0025	0.00051	3.2E-05
Cubic	2.0E-06	9.4E-07	0.00011

Table 2.5 L_2 **-norm for the hole problem**

	Meshless--Case 1			Meshless--Case 2			FEM		
No. of Nodes	336	1271	2806	336	1271	2806	336	1271	2806
Linear	0.0032	0.0022	0.0017	0.0033	0.0024	0.0012	0.0011	0.00028	0.00012
Quadratic	0.0025	0.0021	0.0037	0.0027	0.0042	0.0048	0.0011	1.4E-05	1.2E-05

CHAPTER 3 - Elastoplastic Meshless Integral Method

Manuscript to be submitted to:

Computer Methods in Applied Mechanics and Engineering – 2007

Authors' Names:

Jianfeng Ma, X.J. Xin, and Prakash Krishnaswami

Authors' Affiliations

Department of Mechanical and Nuclear Engineering, Kansas State University, 302 Rathbone Hall, Manhattan, KS 66506-5205

3.1 Abstract

In this paper, the meshless integral method based on regularized boundary integral equation [1] has been extended to elastoplastic materials. In the formulation, the domain of interest is populated with a set of nodes using an automatic node generation algorithm. The sub-domain and support domain associated with each node are also generated automatically using algorithms that have been developed for this purpose. The governing integral equation is obtained from the weak form of elastoplasticity over a local sub-domain, and the moving least-squares approximation is used for meshless function approximation. The constitutive law is the small deformation, rate-independent flow theory based on von Mises yielding criterion with isotropic hardening. The collocation method is employed to enforce the essential boundary conditions exactly, which is simple and computationally efficient. The natural boundary conditions are incorporated in the system governing equation and require no special handling. The solution algorithm for elastoplastic analysis is discussed in detail. The proposed method can handle any prescribed loading profile, including unloading and reversed loading. Numerical examples show that the elastoplastic integral meshless method is accurate and robust.

KEY WORDS: meshless method, local boundary integral equation, moving least-squares approximation, subtraction method, singularity removal, elastoplasticity

3.2 Introduction

Over the past two decades the meshless methods have attracted much attention owing to their advantages in adaptivity, higher degree of continuity in the solution field, and capability to handle moving boundary and changing geometry. In the meshless method, the concept of an element is eliminated. The model geometry consists of a distribution of nodes over the domain and the approximate solution is constructed entirely based on these nodes. Consequently, the nodal connectivity in the meshless method is much more flexible. Each node, through the use of a localized weight function, is always connected to the nodes that are nearby, and there is no limitation on the number of other nodes a node can be connected to directly. This makes the model building in meshless methods much simpler than in FEM: adaptivity for meshless methods is easily achieved by adding or removing nodes without the troublesome remeshing of elements.

Several versions of meshless methods have been developed. The initial idea of meshless methods dates back to the Smoothed Particle Hydrodynamics method (SPH) [2] for modeling astrophysical phenomena in the 1970's. Later developments include Diffuse Element Method (DEM) [3], Element-Free Galerkin method (EFG) [4][5][6][7], Reproducing Kernel Particle Method (RKPM) [8][9][10][11], HP-Meshless Cloud method [12][13], Local Boundary Integral Equation (LBIE) method [1][14][15][16][17], meshless local Petrov-Galerkin (MLPG) method [18][19][20][21], boundary node method (BNM) [22][23], natural element method [24], boundary point interpolation method (BPIM) [25], local point interpolation method (LPIM) [26], point interpolation method [27], point assembly method (PAM) [28], and so on. In DEM [3], the usual FEM interpolation is replaced by a diffuse approximation through the use of a least-squares approximation. The method has been applied to two-dimensional (2D) problems in potential theory and linear elasticity. The main idea of EFG method is to use moving least-squares approximation (MLSA) to construct the trial functions used in the Galerkin weak form. The EFG method was applied to problems in linear elasticity and heat conduction [4], fracture mechanics [29], crack propagation [30], thin plate [6], wave propagation and dynamic fracture [7], elasto-plastic fracture mechanics [31], and shape sensitivity analysis and optimization [32]. The BNM involves a coupling between MLSA and boundary integral equations, and has been used for solving problems in both potential theory and elasticity [22][23].

Most development in meshless methods to date has been focused mainly on linear elastic materials. Research in elastoplasticity using meshless methods has not been wide spread and is only currently gaining attention. In [34], Rao proposed an enriched meshless method for fracture analysis of crack in non-linear-elastic, two dimensional solids, subject to mode-I loading conditions. It involves an element-free Galerkin method and two new enriched basis functions to capture the HRR singularity field in non-linear fracture mechanics [35][36]. In [37], Kargarnovin extended the element-free Galerkin method to elastoplastic stress analysis. By using the incremental formulation of plastic deformation a system of elastoplastic EFGM was derived. In [31] and [38], Xu proposed an element-free Galerkin based formulation for dynamic and steady quasi-static crack growth in elastic-plastic materials undergoing small scale yielding respectively. The experimental curve relating the applied load with the incremental crack size was used as input and the displacement, strains, internal state variables, and plastic energy were calculated as output. In [39], Chen formulated the local and nonlocal field theories using dynamic meshless method and performed stress analysis of two crack problems. In [40], element-free Galerkin-finite element coupling method was used to solve elastoplastic contact problems. In [41], Belinha used element-free Galerkin method to perform elastoplastic analysis of plates.

The methods cited above are “meshless” only in terms of the interpolation or approximation of the field or boundary variables, as compared to the usual boundary element method (BEM) or finite element method (FEM), but still have to use a background mesh to integrate a weak form over the problem domain or the boundary. The requirement of a background mesh for integration makes these methods not truly meshless.

The authors have developed a meshless integral method based on regularized boundary integral equation for linear elasticity [1]. This method is a truly meshless method and does not require a background mesh for integration. In the present work, we extend the meshless integral method based on the regularized boundary integral equation to elastoplasticity. The governing integral equation is obtained from the weak form of elastoplasticity over a local sub-domain, and the moving least-squares approximation is used for meshless function approximation. The constitutive law is the small deformation, rate-independent flow theory based on von Mises yielding criterion. Strain hardening is represented by a general isotropic model. The fixed point iteration is used to solve the nonlinear equations because of elastoplasticity. The elastoplastic

meshless integral method based on the regularized integral equation is accurate and robust, and appears remarkably promising. Because few closed-form solutions for realistic elastoplastic engineering problems are available, we compare the results from the current meshless method with FEM results. For all numerical tests, the numerical results are in excellent or satisfactory agreement with FEM solutions.

The structure of this paper is as follows: in Section 3.3, the regularized local boundary integral equation is derived. The subtraction method is used to remove the strong singularity that is present in the local boundary integral equation. In Section 3.4, elastoplastic constitutive equations are presented. Section 3.5 describes the moving least-squares approximation (MLSA). The meshless implementation of the regularized boundary integral equation using MLSA is presented in Section 3.6. Section 3.7 discusses the treatment of weak singularity along with imposition of essential and natural boundary conditions. The solution algorithm for solving the nonlinear elastoplastic equations is given in Section 3.8. Numerical examples are presented in Section 3.9 to assess the accuracy and effectiveness of the method. Discussion and conclusions from this study are given in Section 3.10.

3.3 Regularized Local Boundary Integral Equation Using Subtraction Method

Consider an elastoplastic body represented by a domain Ω with boundary Γ. The small deformation elastoplasticity governing equations are as follows:

$$\begin{cases} \sigma_{ij,j}(\mathbf{x}) + b_i(\mathbf{x}) = 0 \\ \varepsilon_{ij}(\mathbf{x}) = \dfrac{1}{2}(u_{i,j}(\mathbf{x}) + u_{j,i}(\mathbf{x})) \qquad \forall \mathbf{x} \in \Omega \\ d\sigma_{ij}(\mathbf{x}) = C^{e}_{ijkl} d\varepsilon_{kl}(\mathbf{x}) + d\sigma^{cor}_{ij}(\mathbf{x}) \end{cases} \tag{1}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ (bold face denotes vectors or tensors) are the stress and the strain tensor associated with the displacement $\mathbf{u}$, $\mathbf{b}$ is the body force in the domain Ω, C^{e}_{ijkl} is the elastic stiffness matrix, and $d\sigma^{cor}_{ij}$ is the correction term because of elastoplasticity. An index i following a comma designates partial differentiation with respect to x_i, and repeated indices indicate summation over the dimensionality of the problem. The essential and the natural boundary conditions on the boundary Γ are respectively:

$$u_i = \bar{u}_i \text{ on } \Gamma_u \tag{2}$$

$$t_i = \sigma_{ij} n_j = \bar{t}_i \text{ on } \Gamma_t \tag{3}$$

here, $\bar{\mathbf{u}}$ represents the prescribed displacement on Γ_u, $\bar{\mathbf{t}}$ represents the prescribed traction on Γ_t; $\mathbf{n}$ is the outward unit normal to the boundary; $\Gamma_t \cup \Gamma_u = \Gamma$ and $\Gamma_t \cap \Gamma_u = \varnothing$.

The weak form of (1) over a local domain is:

$$\int_{\Omega^{\langle a \rangle}} \left(\sigma_{ij,j}(\mathbf{x}) + b_i(\mathbf{x})\right) g_i\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) d\Omega(\mathbf{x}) = 0 \tag{4}$$

where the notation $< \cdot >$ in this paper is used to denote a node (e.g., <a> indicates node a, <b> indicates node b, etc.) in order to reserve the usual subscripts, i, j, etc., for denoting degree of freedom (DOF) components, $\Omega^{<a>} \subset \Omega$ is a sub-domain related to node <a>, $\mathbf{y}^{\langle a \rangle}$ is the position vector of node a which is also called a source point, $\mathbf{x}$ is the integration or field point which may or may not coincide with a node, and g_i is the test function. In the following, the functional dependence on $\mathbf{x}$, i.e. "$(\mathbf{x})$", will be dropped for brevity when no ambiguity is caused. In this work, following Atluri et al. [14][15], we use a special test function defined by

$$g_i\left(\mathbf{x}, \mathbf{y}^{<a>}\right) = \tilde{u}^*_{ji}\left(\mathbf{x}, \mathbf{y}^{<a>}\right) e_j\left(\mathbf{y}^{<a>}\right) \tag{5}$$

where e_j represents the j-th component of a unit force vector, $\tilde{u}^*_{ji}$ is the special test function, given by [1]:

$$\tilde{u}^*_{ij}(\mathbf{x}, \mathbf{y}^{<a>}) = \frac{1}{8\pi\mu(1-\bar{\nu})}\left\{\left[(4\bar{\nu}-3)\ln\frac{r^{<a>}}{h_s^{<a>}} - \frac{5-4\bar{\nu}}{2(3-4\bar{\nu})}\left(1-\frac{r^{<a>2}}{h_s^{<a>2}}\right)\right]\delta_{ij} + \left(1-\frac{r^{<a>2}}{h_s^{<a>2}}\right)\frac{r_i^{<a>} r_j^{<a>}}{r^{<a>2}}\right\} \tag{6}$$

The associated traction is:

$$\begin{aligned}\tilde{t}^*_{ij}(\mathbf{x}, \mathbf{y}^{<a>}) = -\frac{1}{4\pi(1-\bar{\nu})}\Bigg\{ &\left[\frac{1}{r^{<a>}}\frac{\partial r^{<a>}}{\partial n^{<a>}}(1-2\bar{\nu}) - \frac{r_k^{<a>} n_k^{<a>}}{(3-4\bar{\nu})h_s^{<a>2}}\right]\delta_{ij} + 2\frac{\partial r^{<a>}}{\partial n^{<a>}}\frac{r_i^{<a>} r_j^{<a>}}{r^{<a>3}} \\ &- \frac{1-2\bar{\nu}}{r^{<a>2}}\left(r_i^{<a>} n_j^{<a>} - r_j^{<a>} n_i^{<a>}\right) + \frac{3 r_i^{<a>} n_j^{<a>} - r_j^{<a>} n_i^{<a>}}{(3-4\bar{\nu})h_s^{<a>2}} \Bigg\}\end{aligned} \tag{7}$$

where $r^{<a>}$ is the distance from $\mathbf{y}^{\langle a \rangle}$ to $\mathbf{x}$; $r_i^{<a>} = x_i - y_i^{<a>}$; $\mathbf{n}^{<a>}$ is the outward unit normal to the boundary $\partial\Omega^{\langle a \rangle}$ at $\mathbf{x}$; $\bar{\nu} = \nu$ and $\bar{E} = E$ for plane strain, or $\bar{\nu} = \nu/(1+\nu)$ and $\bar{E} = E(1-\bar{\nu}^2)$ for

plane stress; E is the Young's modulus, ν is the Poisson's ratio, and $\mu = \overline{E}/\left(2\left(1+\overline{\nu}\right)\right)$ is the shear modulus. $h_s^{<a>}$ is the radius of the local sub-domain.

The special test function $\tilde{u}_{ij}^*$ has the property that it vanishes on the boundary of a spherical $\Omega^{\langle a\rangle}$. With $\tilde{u}_{ij}^*$ as the test function, application of integration by parts to (4) twice leads to [1]:

$$\begin{aligned} &-\int_{\Omega^{<a>}} \delta\left(\mathbf{x}, \mathbf{y}^{<a>}\right) u_i(\mathbf{x}) d\,\Omega(\mathbf{x}) + \int_{\partial\Omega^{<a>}} \tilde{u}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{<a>}\right) t_j(\mathbf{x}) d\,\Gamma(\mathbf{x}) \\ &-\int_{\partial\Omega^{<a>}} \tilde{t}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{<a>}\right) u_j(\mathbf{x}) d\,\Gamma(\mathbf{x}) - \int_{\Omega^{<a>}} \tilde{u}_{ij,k}^*\left(\mathbf{x}, \mathbf{y}^{<a>}\right) \sigma_{ij}^{cor}(\mathbf{x}) d\Omega(\mathbf{x}) \\ &+\int_{\Omega^{<a>}} \tilde{u}_{ij}^*\left(\mathbf{x}, \mathbf{y}^{<a>}\right) b_j(\mathbf{x}) d\,\Omega(\mathbf{x}) = 0 \end{aligned} \tag{8}$$

where $\partial\Omega^{\langle a\rangle}$ is the boundary of $\Omega^{\langle a\rangle}$. Equation (8) in its current form cannot be used directly in numerical calculation because when $y^{<a>}$ is a boundary node, the equation contains strong singularity (1/r type in line integral) in the traction term $\tilde{t}_{ij}^*$. For the local integral equation approach to be a valid numerical method, the strong singularity must be handled appropriately.

The subtraction technique is employed in the present study to remove the strong singularity. The technique has been presented in [1], but sufficient details will still be given here for completeness. For simplicity in implementation, the local sub-domain is always chosen as a sphere or part of a sphere centered on a node. If node $\mathbf{y}^{<a>}$ is an interior node, $h_s^{<a>}$ is selected such that $\Omega^{<a>}$ stays fully inside Ω. If $\mathbf{y}^{<a>}$ is a boundary node, then $\Omega^{<a>}$ is the intersection of Ω and a sphere $\Omega^{'<a>}$ of radius $h_s^{<a>}$, centered at the boundary node, and the boundary $\partial\Omega^{<a>}$ is the union of the part of $\partial\Omega'^{\langle a\rangle}$ inside Ω and the part of $\partial\Omega^{<a>}$ inside $\Omega^{'<a>}$, as illustrated in Figure 3.1. A further modification is the exclusion of a tiny sphere Ω_Δ of radius Δ (which later tends to zero) centered on $\mathbf{y}^{<a>}$. Figure 3.2 shows schematically this modification.

We now decompose the boundary $\partial\Omega^{<a>}$ into the following sections:

$$\partial\Omega^{<a>} = C^{<a>} \cup C_\Delta^{<a>} \cup \Gamma^{<a>} \tag{9}$$

$$\Gamma^{<a>} = \Gamma_u^{<a>} \cup \Gamma_t^{<a>} \tag{10}$$

where $C^{<a>}$ is the circular part of $\partial\Omega^{<a>}$ of radius $h_s^{<a>}$, C_Δ is the circular part of $\partial\Omega_\Delta$ of radius Δ, $\Gamma_u^{<a>}$ is the section of $\Gamma^{<a>}$ where the displacement is prescribed and $\Gamma_t^{<a>}$ the section where the traction is prescribed. For interior nodes, $\Gamma^{<a>}$ is zero, and $C^{<a>}$ is a full circle.

Using subtraction method, we obtain, in the limit of $\Delta \to 0$, the following:

$$\begin{aligned} &- \int_{\Omega^{<a>}} \tilde{u}^*_{ij,k}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)\sigma^{cor}_{ij}(\mathbf{x})d\Omega(\mathbf{x}) + \int_{\Gamma^{<a>}} \tilde{u}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)t_j(x)d\Gamma(\mathbf{x}) \\ &- \int_{C^{<a>}} \tilde{t}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)(u_j(\mathbf{x}) - u_j^{<a>})d\Gamma(\mathbf{x}) \\ &- \int_{\Gamma^{<a>}} \tilde{t}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)(u_j(\mathbf{x}) - u_j^{<a>})d\Gamma(\mathbf{x}) + \int_{\Omega^{<a>}} \tilde{u}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)b_j(\mathbf{x})d\Omega(\mathbf{x}) = 0 \end{aligned} \tag{11}$$

The integration of $\tilde{t}^*_{ij}(\mathbf{x}, \mathbf{y}^{<a>})$ over $C^{<a>}$ can be obtained in closed form [1]:

$$\int_{C^{<a>}} \tilde{t}^*_{ij}(\mathbf{x}, \mathbf{y}^{<a>})d\Gamma = -\alpha_{ij}^{<a>} \tag{12}$$

with

$$-[\alpha_{ij}^{<a>}] = \begin{bmatrix} -\frac{\theta}{2\pi} - \frac{\sin 2\theta_2 - \sin 2\theta_1}{2\pi(3-4\bar{\nu})} & \frac{\cos 2\theta_2 - \cos 2\theta_1}{2\pi(3-4\bar{\nu})} \\ \frac{\cos 2\theta_2 - \cos 2\theta_1}{2\pi(3-4\bar{\nu})} & -\frac{\theta}{2\pi} + \frac{\sin 2\theta_2 - \sin 2\theta_1}{2\pi(3-4\bar{\nu})} \end{bmatrix} \tag{13}$$

Here $\theta_2 - \theta_1 = \theta$ is the internal boundary angle subtended by material at $\mathbf{y}^{\langle a \rangle}$ on the boundary, as shown in Figure 3.3. Two special cases are worth noting. For an interior node, $\theta = 2\pi$; while for a boundary node where the boundary is smooth, $\theta = \pi$.

Substituting the above expressions into (11) leads to:

$$\begin{aligned} \alpha_{ij}^{<a>}u_j^{<a>} &= -\int_{\Omega^{<a>}} \tilde{u}^*_{ij,k}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)\sigma^{cor}_{ij}(\mathbf{x})d\Omega + \int_{\Gamma_u^{<a>}+\Gamma_t^{<a>}} \tilde{u}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)t_j(\mathbf{x})d\Gamma \\ &- \int_{C^{<a>}} \tilde{t}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)u_j(\mathbf{x})d\Gamma - \int_{\Gamma_u^{<a>}+\Gamma_t^{<a>}} \tilde{t}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)(u_j(\mathbf{x}) - u_j^{<a>})d\Gamma \\ &+ \int_{\Omega^{<a>}} \tilde{u}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{<a>}\right)b_j(\mathbf{x})d\Omega \end{aligned} \tag{14}$$

In (14), the subtraction method has been used in the forth term of the right hand side. When the field point $\mathbf{x}$ approaches the source node $\mathbf{y}^{\langle a \rangle}$, $u_i(x) - u_i^{<a>}$ tends to zero which removes the strong singularity and makes the integral numerically integrable. All other terms in

(14) are regular or weakly singular for which special integration quadrature gives convergent and accurate results. The regularized Equation (14) holds for any source node $\mathbf{y}^{\langle a \rangle}$, either inside the domain or on the boundary. In the current meshless integral method, both boundary and interior source nodes are used, and the moving least-squares approximation is employed for approximating the solution field, as presented later.

3.4 Elastoplastic Constitutive Equation

Engineering materials usually exhibit a hardening behavior. Increasing the stress beyond the initial yield surface and into the hardening range generates both plastic and elastic deformation. At each stage of plastic deformation after initial yielding, a new yield surface referred to as subsequent yield surface, is established, i.e. the yield surface is not constant but a function of either plastic work W_p , in the case of working-hardening or plastic strain ε_{ij}^p, in the case of strain hardening. If the state of stress moves inside the new yield surface (unloading), the behavior of the material is elastic and no plastic deformation takes place. The elastoplastic stress-strain behavior is loading path dependent.

Two basic approaches have been used to develop constitutive equations for strain-hardening materials. The first type of formulation is the deformation theory in the form of the total stress-strain relationship which assumes that the state of stress determines the state of strain uniquely as long as the plastic deformation continues. The other type of theory is the incremental theory or flow theory which defines the relationship between the next increment of the plastic strain increment $d\varepsilon_{ij}^p$ and the state of stress σ_{ij} along with the stress increment $d\sigma_{ij}$. This incremental relationship resembles, to some extent, the stress-strain relationship of a viscous fluid and uses the concept of plastic potential function g which defines the ratios of the components of the plastic strain increment. If yield function f coincides with plastic potential function g , the flow rule is called associated type; otherwise it is called nonassociated type.

The yield criterion defines the elastic limits of a material under combined states of stress. There are two commonly used yield criterion: (1) Tresca yield criterion; (2) von Mises yield criterion. The Tresca criterion implies that material yielding occurs when the maximum shear stress reaches a critical value. The von Mises criterion implies that the plastic behavior begins when the octahedral shearing stress reaches a critical value.

Several hardening rules have been proposed to describe the growth of subsequent loading surfaces for strain-hardening materials: (1) isotropic hardening; (2) kinematic hardening; (3) mixed hardening [42].

In this research, we use associated flow theory, von Mises yield criterion, and isotropic strain hardening to model the material behavior.

The von Mises yield criterion has the following form:

$$f = \frac{1}{2} S_{ij} S_{ij} - \kappa = \frac{1}{2} S_{ij} S_{ij} - \frac{1}{3} \sigma_e^{\,2} = 0 \tag{15}$$

f is the yield function, S_{ij} is the component of deviatoric stress defined by $S_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{mm}$, and σ_e is the effective stress, κ is the material parameter. Effective stress $\sigma_e = Y_0 + \varepsilon_p H_p$, Y_0 is the initial yield stress, H_p is the hardening modulus, and ε_p is effective plastic strain. H_p is given by the following equation

$$H_p = \frac{E_T}{1 - E_T/E}, \tag{16}$$

where E_T is elastoplastic tangent modulus and E is the elastic Young's modulus.

The plastic strain increment $d\varepsilon_{ij}^p$ is expressed by an associated flow rule as [42]:

$$d\varepsilon_{ij}^p = \frac{L}{H}\frac{\partial f}{\partial \sigma_{ij}} = \frac{L}{H} S_{ij} \tag{17}$$

where L is the loading criterion and given by the following equation [42]:

$$\begin{aligned} L &= \left\{\frac{\partial f}{\partial \{\sigma\}}\right\}^T [C]\{d\varepsilon\} \\ &= \frac{E}{1+v}[S_{11}d\varepsilon_{11} + S_{22}d\varepsilon_{22} + S_{33}d\varepsilon_{33} + 2S_{12}d\varepsilon_{12}] \end{aligned} \tag{18}$$

[C] is elastic stiffness matrix. This equation is for 2D problems.

The positive scalar H is expressed as [42]:

$$H = \frac{2}{3}\sigma_e^{\,2}\left(\frac{2}{3}\frac{d\sigma_e}{d\varepsilon_p} + \frac{E}{1+v}\right) \tag{19}$$

Using Equation (18) and (19), we obtain the plastic strain increment:

$$\left\{d\varepsilon_{ij}^{\ p}\right\} = \frac{L}{H}\begin{Bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ 2S_{12} \end{Bmatrix} \tag{20}$$

With $d\varepsilon_{ij}^{p}$, and the increment of effective plastic strain is computed as follows:

$$d\varepsilon_p = C\sqrt{d\varepsilon_{ij}^{\ p} d\varepsilon_{ij}^{\ p}} = \sqrt{\frac{2}{3}}\frac{L}{H}\sqrt{S_{ij}S_{ij}} = \sqrt{\frac{2}{3}}\frac{L}{H}\sqrt{\frac{2}{3}}\sigma_{eq} = \frac{2}{3}\sigma_{eq}\frac{L}{H} \tag{21}$$

where $\sigma_{eq} = \sqrt{\frac{3}{2}S_{ij}S_{ij}}$ is the equivalent stress of current stress state.

The constitutive equations are summarized as follows:

a. Elastic region ($f(\sigma_{ij}) < 0$)

$$d\sigma_{ij} = 2\mu d\varepsilon_{ij} + \lambda\frac{d\sigma_{kk}}{2\mu + 3\lambda}\delta_{ij} \tag{22}$$

Where λ and μ are Lame constants.

b. Plastic region (loading, $f(\sigma_{ij}) = 0, L \geq 0$)

$$d\sigma_{ij} = 2\mu d\varepsilon_{ij} + \lambda d\varepsilon_{kk}\delta_{ij} - \frac{9\mu^2}{3\mu + H_p}\frac{S_{kl}d\varepsilon_{kl}S_{ij}}{\sigma_e^2} \tag{23}$$

c. Plastic region (unloading, $f(\sigma_{ij}) = 0, L < 0$)

$$d\sigma_{ij} = 2\mu d\varepsilon_{ij} + \lambda d\varepsilon_{kk}\delta_{ij} \tag{24}$$

3.5 The Moving Least-squares Approximation

In the finite element method, the coupling between the nodes is accomplished through the use of shape functions, defined locally over each element, which interpolate the solution field from nodal values. For a meshless method, the absence of elements excludes the use of such shape functions and therefore, a different local approximation scheme based on nodal values but independent of any elements needs to be devised. In this work, we have chosen to exploit the non-interpolative moving least-squares approximation (MLSA) scheme because of its high accuracy and the ease with which it can be extended to n-dimensional problems.

Consider a domain Ω that contains n nodes:

$$\mathbf{y}^{<a>} = \begin{cases} [y_1^{<a>}, y_2^{<a>}]^T & \text{in 2-D} \\ [y_1^{<a>}, y_2^{<a>}, y_3^{<a>}]^T & \text{in 3-D} \end{cases}, \; a = 1 \cdots n \tag{25}$$

For any of these nodes $\mathbf{y}^{\langle a \rangle}$, following [3][13][14], we define a ***support domain*** for node $\mathbf{y}^{\langle a \rangle}$, which is a sphere (3D) or disk (2D) centered on $\mathbf{y}^{\langle a \rangle}$ with a radius $\ell_w^{\langle a \rangle}$. A weight function $w^{<a>}$ is a continuous function that is positive in the support domain and zero outside, i.e.

$$\begin{cases} w^{\langle a \rangle}(\mathbf{x}) \geq 0 & \text{if } \left| \mathbf{x} - \mathbf{y}^{\langle a \rangle} \right| \leq \ell_w^{\langle a \rangle} \\ w^{\langle a \rangle}(\mathbf{x}) = 0 & \text{if } \left| \mathbf{x} - \mathbf{y}^{\langle a \rangle} \right| > \ell_w^{\langle a \rangle} \end{cases} \tag{26}$$

As introduced previously, the ***sub-domain*** $\Omega^{\langle a \rangle}$ for node $\mathbf{y}^{\langle a \rangle}$, located entirely inside Ω, is a sphere or part of a sphere centered on $\mathbf{y}^{\langle a \rangle}$ with a radius $h_s^{<a>}$. Figure 3.4 illustrates the meaning of local sub-domain and support domain.

Two other frequently used concepts are the domain of definition and the domain of influence. The ***domain of definition*** of a point $\mathbf{x}$ is the set of all nodes whose weight functions are non-zero at $\mathbf{x}$, while the ***domain of influence*** of a node $\mathbf{y}^{<a>}$ is the set of all nodes whose weight functions are non-zero in some part or all of the sub-domain of node $\mathbf{y}^{<a>}$. The domain of definition and the domain of influence are convenient terms in the description of MLSA and local boundary integrals, and are illustrated schematically in Figure 3.5.

The moving least-squares ***approximant*** $\mathbf{u}_h$ to a function **u** is defined by:

$$\mathbf{u}_h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{c}(\mathbf{x}), \; \forall \mathbf{x} \in \Omega \tag{27}$$

The two vectors **p** and **c** are both functions of the spatial coordinates: $\mathbf{x} = [x_1, x_2]^T$ in 2D or $\mathbf{x} = [x_1, x_2, x_3]^T$ in 3D. **p** is a complete monomial basis of m terms (e.g., in 2D, m=3 for a linear basis, and 6 for a quadratic basis), **c** is a coefficient vector which is determined by minimizing a weighted discrete L_2-norm:

$$J(\mathbf{x}) = \sum_{a=1}^{Nx} \left(w^{\langle a \rangle}(\mathbf{x}) \left[\mathbf{p}^T(\mathbf{y}^{\langle a \rangle}) \mathbf{c}(\mathbf{x}) - \hat{\mathbf{u}}^{\langle a \rangle} \right]^2 \right) \tag{28}$$

where $\hat{\mathbf{u}}^{\langle a\rangle}$ is the fictitious nodal displacement that approximates the value of $\mathbf{u}$ at node $\mathbf{y}^{\langle a\rangle}$, and the upper limit of summation, $N_{\mathbf{x}}$, is the total number of nodes in the domain of definition of point $\mathbf{x}$. The matrices $\mathbf{P}$, $\mathbf{W}$ and $\hat{\mathbf{u}}$ are defined by

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^{\mathrm{T}}(\mathbf{y}^{<1>}) \\ \mathbf{p}^{\mathrm{T}}(\mathbf{y}^{<2>}) \\ \cdots \\ \mathbf{p}^{\mathrm{T}}(\mathbf{y}^{<N_x>}) \end{bmatrix}_{N_x \times m} \tag{29}$$

$$\mathbf{W}(\mathbf{x}) = \begin{bmatrix} w^{<1>}(\mathbf{x}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w^{<N_x>}(\mathbf{x}) \end{bmatrix}_{N_x \times N_x} \tag{30}$$

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}^{<1>} \\ \hat{\mathbf{u}}^{<2>} \\ \vdots \\ \hat{\mathbf{u}}^{<N_x>} \end{bmatrix} \tag{31}$$

Minimization of (28) leads to:

$$\mathbf{u}_{\mathrm{h}}(\mathbf{x}) = \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{x})\hat{\mathbf{u}} = \sum_{a=1}^{N_x} \varphi^{<a>}(\mathbf{x})\hat{\mathbf{u}}^{<a>} \tag{32}$$

where:

$$\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}) \tag{33}$$

$$\text{or } \varphi^{<a>}(\mathbf{x}) = \sum_{b=1}^{N_x} p_b(\mathbf{x})\left[\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\right]_{ba} \tag{34}$$

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^{\mathrm{T}}\mathbf{W}(\mathbf{x})\mathbf{P} = \mathbf{B}(\mathbf{x})\mathbf{P} = \sum_{a=1}^{N_x} w^{<a>}(\mathbf{x})\mathbf{p}(\mathbf{y}^{<a>})\mathbf{p}^{\mathrm{T}}(\mathbf{y}^{<a>}) \tag{35}$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{P}^{\mathrm{T}}\mathbf{W}(\mathbf{x}) = \left[w^{<1>}(\mathbf{x})\mathbf{p}(\mathbf{y}^{<1>}) \;\; w^{<2>}(\mathbf{x})\mathbf{p}(\mathbf{y}^{<2>}) \;\; \cdots \;\; w^{<N_x>}(\mathbf{x})\mathbf{p}(\mathbf{y}^{<N_x>})\right] \tag{36}$$

The MLSA for a function exists only when $\mathbf{A}(\mathbf{x})$ is non-singular. A necessary condition for a well-defined MLSA is that for each sample point $\mathbf{x} \in \Omega$ (a node or a quadrature point), at

least *m* weight functions are non-zero and the nodes in the domain of definition of $\mathbf{x}$ are not arranged in a degenerate pattern (such as on a straight line). In MLSA, the shape function related to node $\mathbf{y}^{<a>}$ is $\varphi^{<a>}(\mathbf{x})$. The size of the support domain should be large enough to ensure the coupling between a minimum set of nodes, but small enough to capture local variations.

The influence of the choices of the basis functions and the weight functions on the behavior and the quality of the shape function has been discussed in [1]. In this work, we use linear, quadratic monomial basis, and spline weight functions, defined as follows:

$$\text{Spline:} \qquad w^{\langle a\rangle}(\mathbf{x}) = \begin{cases} 1 - 6\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^2 + 8\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^3 - 3\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^4, & 0 \le r^{\langle a\rangle} \le \ell_w^{\langle a\rangle} \\ 0, & r^{\langle a\rangle} > \ell_w^{\langle a\rangle} \end{cases} \tag{37}$$

here $r^{<a>}$ is the distance from node $\mathbf{y}^{\langle a\rangle}$ to point $\mathbf{x}$, $\ell_w^{\langle a\rangle}$ is the size of support domain. The weight function has only one parameter, the size of support $\ell_w^{\langle a\rangle}$, which makes its use simpler. It is noted that MLSA is non-interpolative, and there is a difference between the nodal value of the MLSA approximant $\mathbf{u}_h$ and the fictitious nodal displacement $\hat{\mathbf{u}}^{<a>}$. For brevity, the subscript h in $\mathbf{u}_h$ will be omitted in the remaining part of this paper.

3.6 Meshless Implementation

We now apply the MLSA to the integral Equation (14) to establish the meshless implementation. The shape function, as we have defined it, gives:

$$u_j(\mathbf{x}) = \sum_{b=1}^{N_x} \varphi^{<b>}(\mathbf{x})\hat{u}_j^{<b>} \tag{38}$$

$$u_{j,k}(\mathbf{x}) = \sum_{b=1}^{N_x} \varphi_{,k}^{<b>}(\mathbf{x})\hat{u}_j^{<b>} \tag{39}$$

where N_x is the total number of nodes in the domain of definition of point $\mathbf{x}$.

The related traction term $t_j(\mathbf{x})$ is

$$t_j(\mathbf{x}) = \sigma_{ij}(\mathbf{x})n_i(\mathbf{x}) \tag{40}$$

where (n_1, n_2) is the normal to the plane passing $\mathbf{x}$ over which the traction acts. For a node $\mathbf{y}^{\langle b\rangle}$, we define $\mathbf{N}$ and $\mathbf{B}^{\langle b\rangle}$ matrices as:

$$\mathbf{N} = \begin{bmatrix} n_1 & 0 & n_2 \\ 0 & n_2 & n_1 \end{bmatrix};\ \mathbf{B}^{<b>} = \begin{bmatrix} \varphi_{,1}^{<b>} & 0 \\ 0 & \varphi_{,2}^{<b>} \\ \varphi_{,2}^{<b>} & \varphi_{,1}^{<b>} \end{bmatrix} \tag{41}$$

Combining (41) with (39) and (40), we can express the traction in terms of the shape functions as follows:

$$\begin{Bmatrix} t_1(\mathbf{x}) \\ \\ t_2(\mathbf{x}) \end{Bmatrix} = \sum_{b=1}^{N_x} [\mathbf{N}\mathbf{C}_{ijkl}^{ep}\mathbf{B}^{<b>}(\mathbf{y}^{<a>})] \begin{Bmatrix} \hat{u}_1^{<b>} \\ \\ \hat{u}_2^{<b>} \end{Bmatrix} \tag{42}$$

here C_{ijkl}^{ep} is the elastoplastic stiffness matrix.

With the above discretization and the boundary conditions that $u_j = \bar{u}_j$ on $\Gamma_u^{<a>}$ and $t_j = \bar{t}_j$ on $\Gamma_t^{<a>}$, Equation (14) becomes (there is a summation on b and j but not on a and i):

$$\sum_{b=1}^{N_y}(H_{ij}^{<a,b>}\hat{u}_j^{<b>}) - \sum_{b=1}^{N_y}(L_{ij}^{<a,b>}\hat{u}_j^{<b>}) + \alpha_{ij}^{<a>}u_j^{<a>} = G_i^{<a>} \tag{43}$$

where $H_{ij}^{\langle a,b\rangle}$, $L_{ij}^{<a,b>}$ and $G_i^{\langle a\rangle}$ are:

$$\begin{aligned} H_{ij}^{<a,b>} = &-\int_{\Gamma_u^{<a>}} \tilde{u}_{ik}^*(\mathbf{x},\mathbf{y}^{<a>})[\mathbf{N}\mathbf{C}^{ep}\mathbf{B}^{<b>}(\mathbf{x})]_{kj}d\Gamma + \int_{C^{<a>}} \tilde{t}_{ij}^*(\mathbf{x},\mathbf{y}^{<a>})\varphi^{<b>}(\mathbf{x})d\Gamma \\ &+ \int_{\Gamma_t^{<a>}} \tilde{t}_{ij}^*(\mathbf{x},\mathbf{y}^{<a>})(\varphi^{<b>}(\mathbf{x}))d\Gamma \end{aligned} \tag{44}$$

$$L_{ij}^{<a,b>} = \int_{\Gamma_t^{<a>}} \tilde{t}_{ij}^*(\mathbf{x},\mathbf{y}^{<a>})(\varphi^{<b>}(\mathbf{y}^{<a>}))d\Gamma \tag{45}$$

$$\begin{aligned} G_i^{<a>} = &\int_{\Omega^{<a>}} b_j(\mathbf{x})\tilde{u}_{ij}^*(\mathbf{x},\mathbf{y}^{<a>})d\Omega + \int_{\Gamma_t^{<a>}} \tilde{u}_{ij}^*(\mathbf{x},\mathbf{y}^{<a>})\bar{t}_j(\mathbf{x})d\Gamma \\ &- \int_{\Omega^{<a>}} \tilde{u}_{ij,k}^*\left(\mathbf{x},\mathbf{y}^{<a>}\right)\sigma_{ij}^{cor}(\mathbf{x})d\Omega - \int_{\Gamma_u^{<a>}} \tilde{t}_{ij}^*(\mathbf{x},\mathbf{y}^{<a>})(\bar{u}_j(\mathbf{x}) - \bar{u}_j(\mathbf{y}^{<a>}))d\Gamma \end{aligned} \tag{46}$$

and the upper limit of summation, N_y, is the total number of nodes in the domain of influence of node $\mathbf{y}^{<a>}$. The third term of Equation (46) is the plastic term because of the elastoplasticity.

In all numerical examples tested in this work, the body force term is zero.

3.7 Treatment for Weak Singularity and Imposition of Boundary Conditions

Owing to the subtraction technique, the singularity in the third integral of (44) as **x** approaches $\mathbf{y}^{<a>}$ is cancelled by the term in (45), and similarly the fourth integral in (46) is also regularized. Even though the subtraction technique removes the strong singularity, the integrands in the first integral of (44) and the second integral in (46) still contain the weakly singular ln(r) term. The logarithmic singularity is integrable, but the accuracy of ordinary Legendre-Gauss integration is poor. We found that the special integration scheme for the logarithmic singularity [43], which is reproduced in Table 3.1 for completeness, achieves excellent numerical accuracy [1]. The remaining integrals in Equation (44) to (46) are all regular for which standard numerical quadrature produces accurate results.

In our numerical examples, the numbers of integration points were as follows: 8 integration points for any integral along a straight line, and 64 integration points for any integral over an sub-domain. For regularized integrals, the usual Legendre-Gauss integration was used. For integrals containing logarithmic singularity, the special integration of 8 integration points listed in Table 3.1 was used.

Appropriate boundary conditions need to be imposed in order to solve the simultaneous Equations (43). In meshless methods, imposing the essential (Dirichlet) boundary conditions is not as easy as in the finite element method. Because MLSA is non-interpolative, the essential boundary condition does not take the form of prescribed value for the fictitious nodal displacement ($\hat{u}_i^{<a>} = \bar{u}_i^{<a>}$), but rather a constraint equation involving a linear combination of the fictitious nodal displacements in a neighborhood of the boundary node (i.e., $u_i^{\langle a\rangle} = \sum_{b=1}^{n} \phi^{\langle b\rangle}\left(\mathbf{y}^{\langle a\rangle}\right)\hat{u}_i^{\langle b\rangle} = \bar{u}_i^{\langle a\rangle}$). A number of techniques for the imposition of essential boundary conditions have been developed, including: (1) Collocation methods [5]; (2) Lagrange multiplier method [4]; (3) Penalty method [44][45]; (4) Nitsche's method [46]; (5) Coupled meshless-finite element method [47]; (6) Admissible approximation method [48]; (7) Method based on d'Alembert principle [49]; (8) Use of window or correction functions that vanish on the boundary [12]; (9) Discrete form of essential boundary conditions [7]; and (10) Displacement constraint

equation method [50]. The advantages and disadvantages of each of these method have been discussed briefly in [1].

Collocation methods for enforcing essential boundary conditions are defined as those methods in which conditions are enforced exactly at a discrete set of boundary nodes [51]. A number of collocation methods have been developed. The direct collocation method [5] used the collocation condition

$$\hat{u}_i^{<a>} = \overline{u}_i^{<a>} \tag{47}$$

to replace the row of the discretized weak form equation corresponding to the degree of freedom with prescribed displacement $\overline{u}_i^{<a>}$. This is actually inconsistent with the assumption of MLSA since the fictitious nodal displacement $\hat{u}_i^{<a>}$ is generally not equal to the approximated displacement value.

A modified collocation method uses

$$u_i^{\langle a \rangle} = \sum_{b=1}^{n} \phi^{\langle b \rangle}\left(\mathbf{y}^{\langle a \rangle}\right)\hat{u}_i^{\langle b \rangle} = \overline{u}_i^{\langle a \rangle} \tag{48}$$

as the collocation condition which was shown to yield more accurate results [50]. Wagner and Liu [51] developed a corrected collocation method which restores the consistency of the weak form and enhances convergence. Wu and Plesha [52] proposed a boundary flux collocation method to enforce the boundary conditions exactly.

Generally, there are two types of discretization in meshless methods: (1) Local collocation over multiple local domains, which is employed in [14][15][16][18][20][21]; (2) Galerkin based method over the global domain, which is used in EFG [4][5][6][7], clouds [12], RKPM [8][9][10][11], etc. For local collocation based discretization, each equation is obtained by applying the weak form over a particular local domain, and the weak form needs to be applied n times for a problem with n DOFs. Consequently, the collocation method with (48) can be used easily and directly to impose essential boundary conditions, because each of the system equations is independent of the rest, and replacing the equation corresponding to a constrained DOF by (48) will not cause any inconsistency in the weak form. For the Galerkin based discretization, the n system equations are obtained by applying the weak form over the global domain once, and therefore all equations must hold simultaneously in order to maintain consistency in the weak form. Replacing a row in the matrix equation by (48), which contains a

linear combination of DOFs rather than dictating the value of the single constrained DOF, sacrifices the consistency of the weak form and compromises the solution accuracy of Galerkin based discretization method.

The current meshless integral method utilizes the local collocation based discretization (over multiple sub-domains), and the collocation method (48) can be directly used to impose essential boundary conditions. Since the system equations are obtained by applying the integral Equation (14) to each source node over a local sub-domain, for a DOF with essential boundary condition, we simply use the essential boundary condition $u_i = \bar{u}_i$ rather than applying the integral Equation (14). This is equivalent to replacing the governing equation corresponding to the DOF with essential boundary condition (48). Our numerical tests presented in the numerical examples section show that the collocation method (48) for imposing essential boundary conditions works very well with the meshless integral method.

For the natural boundary condition $t_i = \bar{t}_i$, no special treatment is needed. The prescribed traction is directly used in the second integral in Equation (46).

After the boundary conditions are imposed, the governing equations can be written as

$$\sum_{b=1}^{N_y} (K_{ij}^{<a,b>} \hat{u}_j^{<b>}) = R_i^{<a>} \tag{49}$$

where

$$K_{ij}^{<a,b>} = \begin{cases} H_{ij}^{<a,b>} - L_{ij}^{<a,b>} + \alpha_{ij}^{<a>} \varphi^{<b>}(\mathbf{y}^{<a>}) & \text{when } u_i^{<a>} \text{ is unconstrained} \\ \varphi^{<b>}(\mathbf{y}^{<a>}) \delta_{ij} & \text{when } u_i^{<a>} = \bar{u}_i^{<a>} \end{cases} \tag{50}$$

$$R_i^{<a>} = \begin{cases} G_i^{<a>} & \text{when } u_i^{<a>} \text{ is unconstrained} \\ \bar{u}_i^{<a>} & \text{when } u_i^{<a>} = \bar{u}_i^{<a>} \end{cases} \tag{51}$$

and the upper limit of summation, N_y, is the total number of nodes in the domain of influence of node $\mathbf{y}^{<a>}$.

3.8 Solution Algorithm for Elastoplasticity

The governing equations are nonlinear due to elastoplasticity. In this work, we will use the fixed point iteration to solve the governing equations which has the following advantages: the derivative of the stiffness matrix is not needed, and the implementation is relatively easy. With

this algorithm, the stiffness matrix is computed only once in the first iteration of the first load increment.

Given an externally applied load, we can use a table to divide the load into several load steps. For each load step, we will do the scaling of the load step. After the scaling, we will divide the remaining part of this load step into several load increments.

Throughout this section, we use $\{\text{var}\}_{[m]}^{(i)}$ to denote a variable $\{\text{var}\}$(displacement, stress or strain) for m-th load increment at ith iteration, and use $\{\text{var}\}_{[m]}$ to denote the converged solution for m-th load increment.

3.8.1 Solution algorithm

1. Divide the given load into N load steps, and set the initial load step n as 1.
2. In current load step, displacement increment or traction over the boundary are specified. Use Equation (50) and (51) to calculate elastic stiffness matrix K and load R , and solve for displacement increment for all nodes.
3. In order to determine the first Gaussian point that enters elastoplastic state, all Gaussian points for all nodes need to be checked.

   3.1. For each Gaussian point, using (52) to compute the strain increment and using (53) to compute the trial stress increment $\{\Delta\sigma^e\}$ assuming elastic behavior:

$$\{\Delta\varepsilon\} = \{\varepsilon\}_{[m+1]}^{(i)} - \{\varepsilon\}_{[m]} \tag{52}$$

$$\{\Delta\sigma^e\} = [C]\{\Delta\varepsilon\} \tag{53}$$

   Where $[C]$ is elastic stiffness matrix. Parameter EPF indicates the stress state at a Gaussian point under consideration. EPF=1 indicates an elastoplastic state, and EPF=0 indicates an elastic state.

   3.2. Determine the loading state

3.2.1. If EFP=1, the Gaussian point was in an elastoplastic state in previous load step. Compute the loading criterion function L using (18). Use parameter r to denote the scaling factor such that:

$$f(\{\sigma\}_{[m]} + r\{\Delta\sigma^e\},\ \kappa_{[m]}) = 0 \tag{54}$$

If L>0, let r=0, plastic loading; If L<=0, let r=1 and EPF=0, compute the yield function after the trial stress increment is applied using $f(\{\sigma\}_{[m]} + r\{\Delta\sigma^e\},\ \kappa_{[m]})$.

If $f \le 0$, let r=1 and EPF=0, the Gaussian point remains in the elastic state; if $f > 0$, let EPF=1, the Gaussian point enters into plastic state and determine the scaling factor r using (54).

3.2.2. If EPF=0, the Gaussian point was in an elastic state in the previous load step. Compute the yield function after the trial stress increment is applied:

$$f(\{\sigma\}_{[m]} + r\{\Delta\sigma^e\},\ \kappa_{[m]})$$

If $f \le 0$, r=1, the Gaussian point remains in the elastic state; if $f > 0$, EPF=1, the Gaussian point enters into plastic state. Determine r such that (54) holds.

3.3. Go through all Gaussian points of all nodes to find the minimal scaling factor that satisfies $0 < r \le 1$. Designate it as r_{min}.

4. Multiply loads, the nodal displacement-increments, stress-increments and strain-increments for all Gaussian points of all nodes by r_{min}, then add results to the present displacements, stresses, and strains. Update them as $\{R\}_{[1]}$, $\{U\}_{[1]}$, $\{\sigma\}_{[1]}$, and $\{\varepsilon\}_{[1]}$.

5. Given the load increment number M and maximum iteration number Imax, initialize both load increment index m and iteration index i as 1.

6. For the current load increment index m and iteration index i, use Equation (55)-(59) to update the solutions.

$$\{U\}_{[m+1]}^{(0)} = \{U\}_{[m]} \tag{55}$$

$$\{F\}_{[m+1]}^{(i-1)} = K\{U\}_{[m+1]}^{(i-1)} - EP\{U\}_{[m+1]}^{(i-1)} \tag{56}$$

$$\{R\}_{[m+1]} = \{R\}_{[m]} + \{\Delta R\} \tag{57}$$

$$[K]\{\Delta U\}^{\{i\}} = \{R\}_{[m+1]} - \{F\}_{[m+1]}^{(i-1)} \tag{58}$$

$$\{U\}_{[m+1]}^{(i)} = \{U\}_{[m+1]}^{(i-1)} + \{\Delta U\}^{(i)} \tag{59}$$

Where $EP\{U\}_{[m+1]}^{(i-1)}$ is the elastoplastic term (the third term of the Equation (46)) of the (m+1)th load increment and (i-1)th iteration.

7. The following convergence criteria is used to terminate the iteration for each load step:

$$\left\|\{R\}_{(m+1)} - \{F\}_{[m+1]}^{(i)}\right\|_2 \le \varepsilon_f \left\|\{R\}_{[m+1]}\right\|_2 \tag{60}$$

ε_f is a predefined tolerance, usually in the order of $10^{-4} \sim 10^{-7}$.

If the program converges, updating the displacement by $\{U\}_{[m+1]} = \{U\}_{[m+1]}^{(i)}$ and go to step 8; otherwise, increase the iteration index by one and check if i is greater than Imax. If i is greater than Imax, the program cannot converge for the preset Imax, exit program execution; otherwise, compute the stresses at all Gaussian points for all nodes and go to step 6.

8. Increase the load increment index m by one and check if m is greater than load increment number M. If Yes, go to step 9; otherwise, go to step 6.
9. Update the load step index n by increasing one and check if n is greater than load step number N. If Yes, exit; otherwise, go to step 2.

As shown in the previous section, when calculating the internal force $\{F\}_{[m+1]}^{(i-1)}$, we need to calculate the $EP\{U\}_{[m+1]}^{(i-1)}$. The integration is usually performed using Gaussian numerical integration techniques. Therefore, the stress state (along with the stress correction term) is computed at all Gaussian points in each iteration step. The following section describes the algorithm for computing the stresses at each Gaussian point.

3.8.2 The procedure used to compute the stress at each Gaussian point

1. Assume the solution is converged for m-th load increment, i.e, for each Gaussian point $\{\varepsilon\}_{[m]}$ and $\{\sigma^{cor}\}_{[m]}$ are known. The strain increment $\{\Delta\varepsilon\}$ and stress increment $\{\Delta\sigma^e\}$ are initially predicted assuming elastic behavior using (52) and (53) and EPF is used to indicate the stress state.
2. Determine the loading state

   2.1 If EFP=1, the Gaussian point was in an elastoplastic state in previous load increment. Compute the loading criterion function L using Equation (18).

If L>0, r=0, plastic loading; if L<=0, let r=1, EPF=0, compute the yield function after the trial stress increment is applied

$$f(\{\sigma\}_{[m]} + \{\Delta\sigma^e\},\ \kappa_{[m]})$$

If $f \le 0$, r=1, the point remains in the elastic state; if $f > 0$, EPF=1, the point enters into elastoplastic state. Determine the scaling factor r such that (54) holds. Update the stress at this Gaussian point by

$$\{\sigma\} = \{\sigma\}_{[m]} + r\{\Delta\sigma^e\} \tag{61}$$

2.2 If EPF=0, the Gaussian point is in an elastic state in the previous load increment. Compute the yield function after the trial stress increment is applied

$$f(\{\sigma\}_{[m]} + \{\Delta\sigma^e\},\ \kappa_{[m]})$$

If $f \le 0$, r=1, the point remains in the elastic state; if $f > 0$, EPF=1, the point enters into elastic state. Determine the scaling factor r such that (54) holds. Update the stress at this point using Equation (61).

3. Given an integer N, compute the sub-increment of strain $\{\Delta\tilde{\varepsilon}\}$, which is defined by (62).

$$\{\Delta\tilde{\varepsilon}\} = \frac{(1-r)\{\Delta\varepsilon\}}{N} \tag{62}$$

Integrate numerically to compute sub-increment of stress $\{\Delta\tilde{\sigma}_{ij}\}$ with n looping from 1 to N, ${}^{(0)}\{\Delta\sigma\} = \{\Delta\sigma^{cor}\} = 0$:

$$\Delta\tilde{\sigma}_{ij} = 2\mu\Delta\tilde{\varepsilon}_{ij} + \lambda\Delta\tilde{\varepsilon}_{kk}\delta_{ij} - \frac{9\mu^2}{3\mu + H_p}\frac{S_{kl}\Delta\tilde{\varepsilon}_{kk}S_{ij}}{\sigma_e^2} \tag{63}$$

$${}^{(n)}\{\Delta\sigma\} = {}^{(n-1)}\{\Delta\sigma\} + \{\Delta\tilde{\sigma}\} \tag{64}$$

4. Update the variables:

$$\{\Delta\sigma^{cor}\} = {}^{(N)}\{\Delta\sigma\} - (1-r)\{\Delta\sigma^e\} \tag{65}$$

$$\{\sigma^{cor}\}^{[i]}_{[m+1]} = \{\sigma^{corr}\}_{[m]} + \{\Delta\sigma^{cor}\} \tag{66}$$

$$\{\sigma\}^{(i)}_{[m+1]} = \{\sigma\} + {}^{(N)}\{\Delta\sigma\} \tag{67}$$

3.8.3 The computation of plastic term

In order to calculate the plastic term in Equation (46), we also need to compute $\tilde{u}^{*}_{ij,k}\left(\mathbf{x},\mathbf{y}^{<a>}\right)$, which is the derivative of $\tilde{u}^{*}_{ij}(\mathbf{x},\mathbf{y}^{<a>})$ with respect to x_k. We use (66) to calculate $\{\sigma^{cor}\}$ for each Gaussian point, and compute the plastic term in Equation (46) accordingly.

3.9 Numerical Examples

This section presents the numerical solutions to several examples using the elastoplastic meshless integral method developed in this study. The examples include the patch tests, shear tests, a finite plate with a circular hole, and a thick-walled cylinder. Because closed form solutions for elastoplastic boundary value problems are few in the literature, except for the simplest cases where closed form solutions are available, we rely on finite element results obtained using the commercial software, ANSYS, as the bases for comparison.

There are two geometric parameters for each node: the size of sub-domain $h_s^{\langle a\rangle}$ and the size of support domain (or the size of weight function) $\ell_w^{<a>}$. The guidelines recommended for the selection of these parameters are given in detail in [1]. A global error indicator, the L_2-norm error in displacement, is defined by

$$L_2\text{-norm error} = \frac{\left\{\sum_{j=1}^{N}\left\{\left(u_1^{\langle j\rangle}\Big|^{num} - u_1^{\langle j\rangle}\Big|^{exact}\right)^2 + \left(u_2^{\langle j\rangle}\Big|^{num} - u_2^{\langle j\rangle}\Big|^{exact}\right)^2\right\}\right\}^{1/2}}{\left\{\sum_{j=1}^{N}\left\{\left(u_1^{\langle j\rangle}\Big|^{exact}\right)^2 + \left(u_2^{\langle j\rangle}\Big|^{exact}\right)^2\right\}\right\}^{1/2}} \times 100\% \qquad (68)$$

In this work, we will use the global L_2-norm error as a measure of the overall performance of the numerical method.

3.9.1 Constant Stress Patch Tests

The first patch test is a 1×1 m^2 square plate shown in Figure 3.6. The material is AISI 1020 steel and the material properties are E=203000 MPa, ν=0.3 , yield stress=260 MPa, the elastoplastic tangent modulus is 1000 MPA. Three meshless models, which are shown in Figure 3.6, were tested. The first is with 9 regular nodes, the second with 25 regular nodes, and the third with 25 irregular nodes. Both plane strain condition and plane stress condition were tested. For these two cases, the left edge is constrained from moving in x1 direction but is free of traction in

x2 direction. The bottom edge is constrained from moving in x2 direction but is free of traction in x1 direction. The prescribed displacement (0.01) in x1 is applied to the right edge which is free of traction in x2 direction. The nature boundary conditions (free traction) are prescribed on the top edge. The spline weight function is used in both cases. The L_2-norm errors for all models are reported in Table 3.2. For plane strain condition, the L_2-norm errors in displacement for the 9 node, 25 node regular and 25 node irregular models 0.0024, 0.0027 and 0.0030 respectively. For plane stress condition, the corresponding L_2-norm errors are 2.6E-5, 2.4E-5, and 3.3E-5 respectively. The results indicate that the method successfully passes all three patch tests. Figure 3.7 and 3.8 show the undeformed model (diamond), deformed meshless model (square), and FEM model (triangle) for 9 node model in plane strain condition and plane stress condition respectively.

Figure 3.9a shows the stress distribution of σ_{11} for 9 node model using a pre- and post-processor we have developed for the meshless integral method. As can be seen in the plot, the meshless results are identical with the hand calculation results. Figure 3.9b shows the status of each Gauss integration point (plus sign means plastic state).

We also tested the performance under cyclic loading for these three meshless models. In the testing, we first loaded the structure until the material yields, then we reverse-loaded the structure until opposite yield stress was reached; then loaded the structure again until the final load was reached. The L_2-norm errors for all models are reported in Table 3.3. For plane strain condition, the L_2-norm errors in displacement for the 9 node, 25 node regular and 25 node irregular models 0.0027, 0.0021 and 0.0038 respectively. For plane stress condition, the corresponding L_2-norm errors are 3.7E-5, 4.3E-5, and 2.6E-5 respectively. Figure 3.10 shows the change of σ_{11} along with the increase of the load and we find that meshless results follow closely the FEM results.

The test results above show that the method successfully passes all patch tests.

3.9.2 Shear Patch Tests

The second patch test is under shear shown in Figure 3.11a. The material is AISI 1020 steel and the material properties are E=203000 MPa, ν=0.3, yield stress=260 MPa, the elastoplastic tangent modulus is 1000 MPA. Three meshless models, which are shown in Figure

3.11, were tested. The first is with 9 regular nodes, the second with 25 regular nodes, and the third with 25 irregular nodes. Both plane strain condition and plane stress condition were tested. For all three models, prescribed displacements are applied to all edges of the plate. The spline weight function was used in these tests. The L_2-norm errors for all models are reported in Table 3.4. All errors are virtually within machine accuracy. Figure 3.12 and Figure 3.13 show the undeformed model (diamond), deformed meshless model (square), FEM model (triangle) for the model with 25 irregular nodes in plane strain condition and plane stress condition respectively. The shear stress is $\sigma_{12} = 156.14$ MPA for both plane stress and plain strain which is identical with FEM results. The results indicate that the method successfully passes all three shear patch tests within machine accuracy.

Figure 3.14a shows the stress distribution of σ_{12} for 9 node model. As can be seen in the plot, the meshless results are identical with the FEM results. Figure 3.14b shows the status of each Gauss integration point (plus sign means plastic state). Figure 3.15 shows the change of σ_{12} along with the increase of shear strain γ and we find that the meshless results match FEM results perfectly.

3.9.3 Finite Plate with a Circular Hole

The next test example is a finite plate with a circular hole in the center subjected to a uniform displacement in x2 direction at the two sides of the plate. Because of symmetry, only the upper right quadrant of the plate was modeled (see Figure 3.16). The actual model used in the simulation is a finite sized plate of 5 m by 5 m with a quarter of a circular hole of radius of 1 m centered at the lower left corner. The model geometry has 336 nodes with nodes concentrated around the hole. The material is ASTM A514 structural steel and the material properties are E=210000 MPA and ν=0.3 and Et=1000 MPA, and yield stress=900 MPA m are used for simulation, and plane stress condition is enforced. The left edge is constrained from moving in x1 direction but is free of traction in x2 direction. The bottom edge is constrained from moving in x2 direction but is free of traction in x1 direction. The prescribed displacement (0.01 m) is applied on the top edge and is free of traction in x1 direction. On the inner circle and right edge, the nature boundary conditions (free traction) are applied.

The spline weight function was used in the calculation, and different $\ell_w^{<a>}$'s were assigned for different nodes. Generally, smaller $\ell_w^{<a>}$ is used for regions with denser nodal distribution.

We first apply the load Uy=0.01 on the upper edge for simulation. Figure 3.17 shows the deformed meshless model (squares), using spline weight function and linear basis. The meshless results are in good agreement with FEM results (triangles), and the L_2-norm errors are 0.0010. The critical displacement of upper edge, b, that causes initial yielding at the hole edge is equal to 0.007672 m.

Figure 3.18 and Figure 3.19 present the distribution of σ_{22} and von Mises stresses along x2=0 of the 336 node model respectively. From these figures we can see that the numerical results are in reasonable agreement with the FEM results.

Figure 3.20 shows the undeformed model (diamond), deformed meshless model when we increase the load with Uy=0.014 m, and FEM model (triangle). The meshless results are in good agreement with FEM results, and the L_2-norm between meshless solutions and FEM solutions is 0.0011. Figures 3.21 and 3.22 present the distribution of σ_{22} and von Mises stress along x2=0 of the 336 node model respectively, and the good agreement between meshless results and FEM is indicated.

Figure 3.23 shows the undeformed model (diamond), deformed meshless model when we increase the load with Uy=0.018 m, and FEM model (triangle). The meshless results match FEM results very well and the L_2-norm between meshless solutions and FEM solutions is 0.0031. Figures 3.24-3.25 present the distribution of σ_{22} and von Mises stress along x2=0 of the 336 node model respectively, and show that the meshless results and FEM solution are comparable with each other.

From Figure 3.17 -Figure 3.25, we find that our meshless results follow closely the FEM results.

Figure 3.26 shows the spreading of the plastic zone when the load (displacement) is increased. For 3.26a, Figure 3.26b, and 3.26c, the corresponding displacements applied on the upper edge are 0.01 m, 0.014 m, and 0.018 m respectively. The parameter b, as mentioned before, is the critical displacement of upper edge that causes initial yielding.

3.9.4 Thick-walled Cylinder

The last example is a thick-walled cylinder subjected to a gradually increasing internal pressure P_i. Because of symmetry, only the upper right quadrant of the plate was modeled (see Figure 3.27a and 3.27b). The model geometry has 651 nodes with nodes concentrated around the inner circle and is used through this section. The material is AISI 1045 unalloyed carbon steel and the material properties are E=215000 MPA, ν=0.3, Et=0 MPA which corresponding to perfect plasticity, and yield stress=400 MPA. Plane strain condition is enforced. The left edge is constrained from moving in x1 direction but is traction free in x2 direction. The bottom edge is constrained from moving in x2 direction but is traction free in x1 direction. On the inner circle, the nature boundary condition (pressure) is applied while the outer circle is free of traction. The spline weight function and linear monomial basis are used in the simulation. The internal pressure changes from 0 to 445 MPA. The initial yielding pressure P_0 is 192 MPA.

Figure 3.28 compares the effect of the sub-domain radius on the numerical results for linear elasticity in which the yield stress is ignored. The minimum nodal distance (minDist) for a node is defined as the minimum distance between this node and all neighbouring nodes. In case 1, we assign the sub-domain radius of each node its minimum distance (minDist). In case 2, for each internal node, we assign the sub-domain radius of this node twice the minimum nodal distance, but for each boundary node, we still use the minDist as the sub-domain radius. In case 3, for each node (both internal nodes and boundary nodes), we assign the sub-domain radius twice the minimum nodal distance. The L_2-norm errors for these three cases are 0.068, 0.019, and 0.0045 respectively. It is clear that the sub-domain radius has a strong influence on the accuracy of the numerical results.

Figure 3.29 shows the undeformed model (diamond), deformed meshless model using case 3 to set the sub-domain radius for each node (square), FEM model (triangle), and analytical solutions (cross) for elastic case. The L_2-norm error between meshless solutions and FEM solutions is 0.0045. The L_2-norm error between FEM solutions and analytical solutions is 8.7E-6. This shows that FEM results and analytical results are practically identical to each other.

Figure 3.30 shows the undeformed model (diamond), deformed meshless model using case 3 to set the sub-domain radius for each node (square), FEM model (triangle), and analytical solutions using Tresca yield criterion (cross) for elastoplastic case with 216 MPA internal

pressure. The L_2-norm error between meshless solutions and FEM solutions is 0.0023. The performance of the 651 node model is comparable with that of FEM model with the same number of nodes.

Figures 3.31a-3.31c present the distribution of σ_{11} , σ_{22} , and von Mises stress along x1=0 of the 651 node model respectively. From the figures we can see that the meshless numerical results are in excellent agreement with the FEM results.

Figure 3.32 shows the undeformed model (diamond), deformed meshless model with 319 MPA internal pressure, and FEM model (triangle). The L_2-norm between meshless solutions and FEM solutions is 0.0036. Figures 3.33a-3.33c present the distribution of σ_{11}, σ_{22}, and von Mises stress along x1=0 of the model respectively. The performance of the 651 node model is comparable with that of FEM model with the same number of nodes.

Figure 3.34 shows the undeformed model (diamond), deformed meshless model with 345 MPA internal pressure, and FEM model (triangle). The L_2-norm between meshless solutions and FEM solutions is 0.0072. Figures 3.35a-3.35b present the distribution of σ_{11} and von Mises stress along x1=0 of the model respectively, and a good agreement between meshless results and FEM is indicated

Figure 3.36 shows the undeformed model (diamond), deformed meshless model with 445 MPA internal pressure, and FEM model (triangle). The L_2-norm between meshless solutions and FEM solutions is 0.0099. Figures 3.37a-3.37b present the distribution of σ_{11} and von Mises stress, respectively, along x1=0 of the model. We can see that the meshless numerical results are in reasonable agreement with the FEM results.

Figures 3.38a-3.38d show the spreading of the plastic zone when the load (internal pressure) is increased. For Figures 3.30a to 3.30d, the internal pressures applied on the inner circle are 261 MPA, 319 MPA, 345 MPA, and 445 MPA respectively.

3.10 Concluding Remarks

In this paper, the meshless integral method based on regularized boundary integral equation [1] has been extended to elastoplasticity. In our formulation, the domain of interest is populated with a set of nodes. The governing integral equation is obtained from the weak form of elastoplasticity over a local sub-domain, and the moving least-squares approximation is used for

meshless function approximation. The collocation method is employed to enforce the essential boundary conditions exactly, which is simple and computationally efficient. The natural boundary conditions are incorporated in the system governing equation and require no special handling. The constitutive law is the small deformation, rate-independent flow theory based on von Mises yielding criterion. Strain hardening is represented by a general isotropic model. The governing equations are solved using fixed point iteration. The proposed method can handle any prescribed loading profile, including unloading and reversed loading. Numerical examples show that the elastoplastic integral method is accurate and robust.

3.11 References

[1] Anthony Bodin, Jianfeng Ma, X.J. Xin, and Prakash Krishnaswami. A meshless integral method based on regularized boundary integral equation. Computer Methods in Applied Mechanics and Engineering. 2006; 195: 6258-6286.

[2] Gingold RA, Moraghan JJ, Smoothed particle hydrodynamics: theory and applications to non-spherical stars, Monthly Notices of the Astronomical Society 1977, 181, 375-389

[3] Nayroles, B., Touzot and G., Villon, P. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational Mechanics. 1992; 10: 307-318.

[4] Belytschko, T., Lu, Y.Y. and Gu, L. Element free Galerkin method. International Journal for Numerical Methods in Engineering. 1994; 37: 229-256.

[5] Lu, YY, Belytschko T, and Gu, L, A new implementation of the element free Galerkin method, Computer Methods in Applied Mechanics and Engineering, 1994, 113, pp.397-414

[6] Krysl P, Belytschko T., Analysis of thin plates by the element-free Galerkin method, Computational Mechanics, 1998, 21, 211-222

[7] Lu YY, Belytschko T, Tabbara M. Element-free Galerkin method for wave propagation and dynamic fracture, Computer Methods in Applied Mechanics and Engineering, 1995, 126, pp.131-153

[8] Liu, W.K. and Chen. Y. Wavelet and multiple scale reproducing kernel methods. International Journal for Numerical Methods in Fluids. 1995; 21: 901-931.

[9] Aluru, N.R. A reproducing kernel particle method for meshless analysis of microelectromechanical systems. Computational Mechanics. 1999; 23: 324-338

[10] Chen, J.S., Pan, C., and Wu, C.T. A Lagrangian reproducing kernel particle method for metal forming analysis. Computational Mechanics. 1998; 22: 289-338.

[11] Chen, J.S., Pan, C., Wu, C.T., and Liu, W.K. Reproducing kernel particle methods for large deformation analysis of non-linear structures. Comput. Methods Appl. Mech. Engrg. 1996; 139 (1-4): 195-227.

[12] Duarte CAM and Oden JT, An h-p adaptive method using clouds, Comput. Methods Appl. Mech. Engrg, 1996, 139, 237-262

[13] Liszka TJ, Duarte CAM, Tworzydlo WW, hp-meshless cloud method, Computer Methods in Applied Mechanics and Engineering, 1996, 139, 263-288

[14] Zhu T., Zhang J-D., and Atluri S.N. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Computational Mechanics. 1998; 21: 223-235.

[15] Atluri S.N., Sladeck J., Sladeck V., and Zhu T. The local boundary integral equation (LBIE) and its meshless implementation for linear elasticity. Computational Mechanics. 2000; 25: 180-198.

[16] Sladek, J., Sladeck, V., and Van Keer, R. Meshless local boundary integral equation for 2D elastodynamic problems. International Journal for Numerical Methods in Engineering. 2003: 235-249.

[17] Long, S. and Zhang. Q. Analysis of thin plates by the local boundary integral equation (LBIE) method. Engineering Analysis with Boundary Elements. 2002; 26: 707-718.

[18] Atluri, S.N. and Zhu, T. A new meshless Local Petrov-Galerkin approach in computational mechanics. Computational Mechanics. 1998; 22: 117-127.

[19] Atluri, S.N. and Zhu, T. New concepts in meshless methods. International Journal for Numerical Methods in Engineering. 2000; 47: 537-556.

[20] Atluri, S.N. and Zhu, T. The Meshless Local Petrov-Galerkin approach for solving problems in elasto-statics. Computational Mechanics. 2000; 225: 169-179.

[21] Liu, G.R. and Gu, Y.T. Meshless local Petrov-Galerkin (MLPG) method in combination with finite element and boundary element approaches. Computational Mechanics. 2000; 26(6):536-546.

[22] Mukherjee, Y.X. and Mukherjee, S. Boundary node method for potential problems. International Journal for Numerical Methods in Engineering. 1997; 40: 797-815.

[23] Chati, M.K. Mukherjee, S and Mukherjee, YX. The boundary node method for three-dimensional linear elasticity. International Journal for Numerical Methods in Engineering. 1999; 46: 1163-1184.

[24] Sukumar N, Moran B, Belytschko T, The natural element method, International Journal for Numerical Methods in Engineering, 1998, 43, 829-887

[25] Gu, Y.T. and Liu, G.R. A boundary point interpolation method for stress analysis of solids. Computational Mechanics. 2002; 28: 47-54.

[26] Liu, G.R. and Gu, Y.T. A local point interpolation method for 2D solid. International J. Struct. Eng. Mech. 2001;11(2):221-236

[27] Liu, G.R. and Gu, Y.T. A point interpolation method for 2D solid. International Journal for Numerical Methods in Engineering. 2001; 50: 937-951.

[28] Liu, G.R. and Yan, L. A study on numerical integration in element-free methods. Proceeding of 4th Asia-Pacific Conference on Computational Mechanics, Singapore.1999: 979-984.

[29] Sukumar N, Moran B, Black T, Belytschko T, An element-free Galerkin method for three-dimensional fracture mechanics, Computational Mechanics, 1997, 20, 170-175

[30] Lee, Sang-Ho and Yoon, Yong-Cheol. Numerical prediction of crack propagation by an enhanced element-free Galerkin method. Nuclear Engineering and Design.2004; 227:257-271.

[31] Xu Y, Saigal S, An element-free Galerkin analysis of steady dynamic growth of a mode I crack in elastic-plastic materials, International Journal of Solids and Structures, 1999, 36, 1045-1079

[32] Bobaru F, Mukherjee S, Shape sensitivity analysis and shape optimization using the element-free Galerkin method, Computer Methods in Applied Mechanics and Engineering, 2001, 190, 4319-4337

[33] Sladek V., Sladek J., Atluri S.N., and Van Keer R. Numerical integration of singularities in meshless implementation of local boundary integral equations. Computational Mechanics. 2000; 25: 394-403.

[34] Rao, B.N. and Rahman, S. An enriched meshless method for non-linear fracture mechanics. International Journal for Numerical methods in Engineerig. 2004; 59: 197-223.

[35] Hutchinson, J.W. Singular behaviour at the end of a tensile crack in a hardening material. Journal of Mechanics and Physics of Solids. 1968; 16: 13-31.

[36] Rice, J.R. and Rosengren, G.F. Plane strain deforming near a crack tip in a power hardening material. Journal of Mechanics and Physics of Solids. 1968; 16: 1-13.

[37] Kargarnovin, M.H., Toussi, H.E., and Fariborz, S.J. Elasto-plastic element-free Galerkin method. Computational Mechanics. 2004; 33: 206-214.

[38] Xu, Y. and Saigal, S. Element free Galerkin study of steady quasi-static crack growth in plane strain tension in elastic-plastic materials. Computational Mechanics. 1998; 22: 206-214.

[39] Chen, Y.P., Lee, J.D., and Eskandarian, A. Dynamic meshless method applied to nonlocal crack problems. Theoretical and Applied Mechanics. 2002: 38: 293-300.

[40] Liu, T., Liu, G., and Wang Q. An elemen-free Galerkin-finite element coupling method for elastoplastic contact problem. Journal of Tribology. 2006; 128: 1-9.

[41] Belinha, and Dinis, L. M. Elastoplastic analysis of plates by the element free Galerkin method. International Journal for Computer-Aided Engineering and Software. 2006; 23(5):525-551.

[42] Chen, W.F. and Han, D. J. *Plasticity for Structural Engineers*. Springer-Verlag Press, 1988.

[43] Stroud, A.H. and Secrest, D. *Gaussian Quadrature Formulas*. Prentice-Hall, 1966.

[44] Zhu, T. and Atluri, S.N. A modified collocation method and a penalty foumulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput. Mech. 1998; 21(3): 211-222.

[45] Gavete, L., Benito, J.J., Falcon S. and Ruiz A. Implementation of essential boundary conditions in a meshless method. *Commun. Numer. Meth. Engng.* 2000; 16: 409-421.

[46] Arnold DN, Brezzi F, Cockburn B, Marini LD, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIMA J. Numer. Anal., 2002, 39, 1749-1779

[47] Hegen D., Element-free Galerkin methods in combination with finite element approaches, Computer Methods in Applied Mechanics and Engineering, 1996, 19, 120-135

[48] Gosz, J. and Liu, W.K. Admissible approximations for essential boundary conditions in the reproducing kernel particle method. Comput. Mech. 1996; 19(2):120-135

[49] Gunther, F. C. and Liu, W. K. Implementation of boundary conditions for meshless methods. Computer Methods in Applied Mechanics and Engineering 1998; 163: 205-230.

[50] Zhang X, Liu X, Lu MW and Chen Y, Imposition of essential boundary conditions by displacement constraint equations in meshless methods, Communications in Numerical Methods in Engineering, 2001, 17, 165-178

[51] Wagner, GJ and Liu WK, Application of essential boundary conditions in mesh-free methods: a corrected collocation method, International Journal for Numerical Methods in Engineering, 2000, 47, 1367-1379

[52] Wu CC and Plesha ME, Essential boundary condition enforcement in meshless methods: boundary flux collocation method, Int. J. Numer. Meth. Engng, 2002, 53, 499-514

[53] Timoshenko, S.P. and Goodier, J.N. *Theory of elasticity*. McGraw-Hill Book Company, 1970.

[54] ANSYS 5.6 Documentation, ANSYS Incorporated, 1999.

Figure 3.1 Schematic diagram showing the sub-domain for an interior or a boundary node $\mathbf{y}^{\langle a \rangle}$

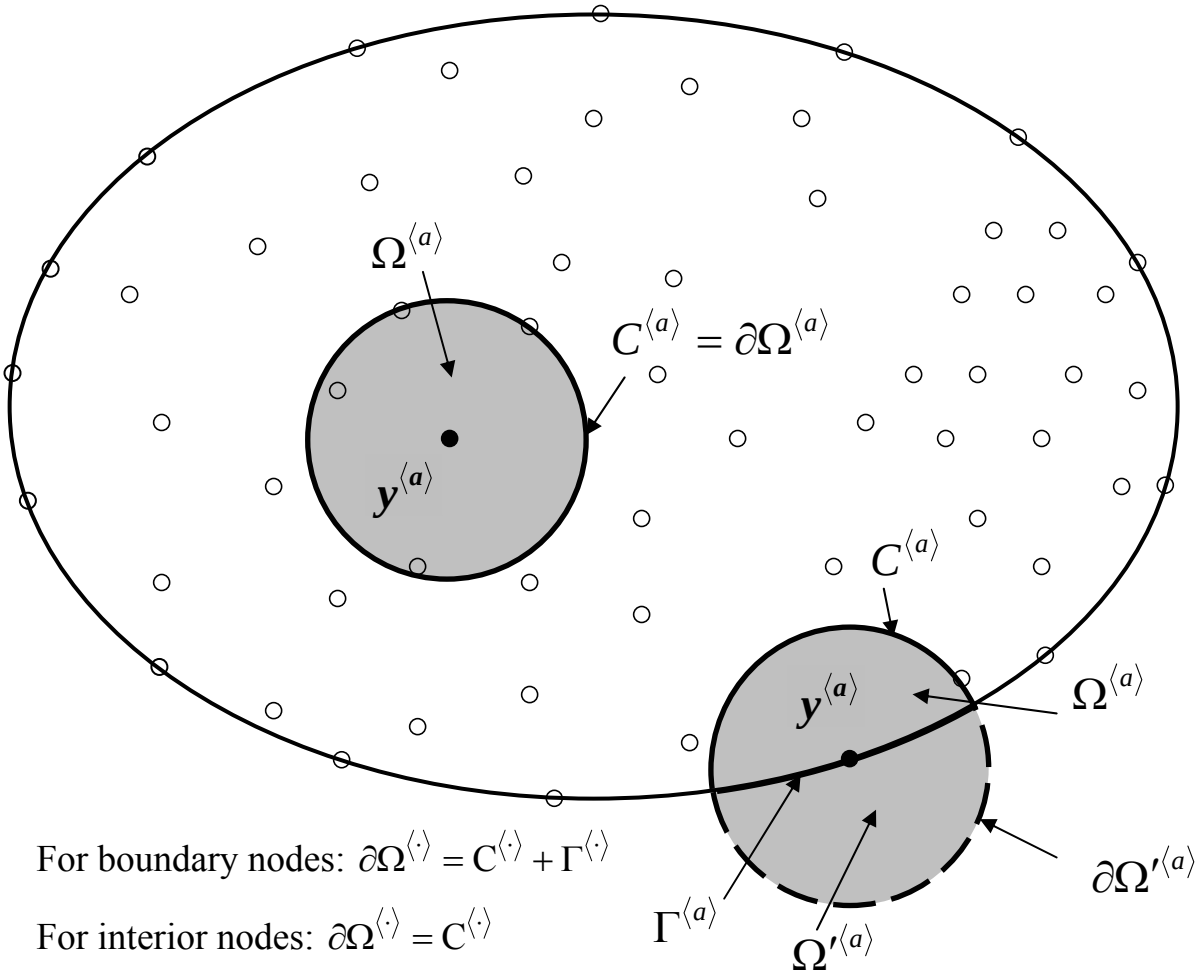
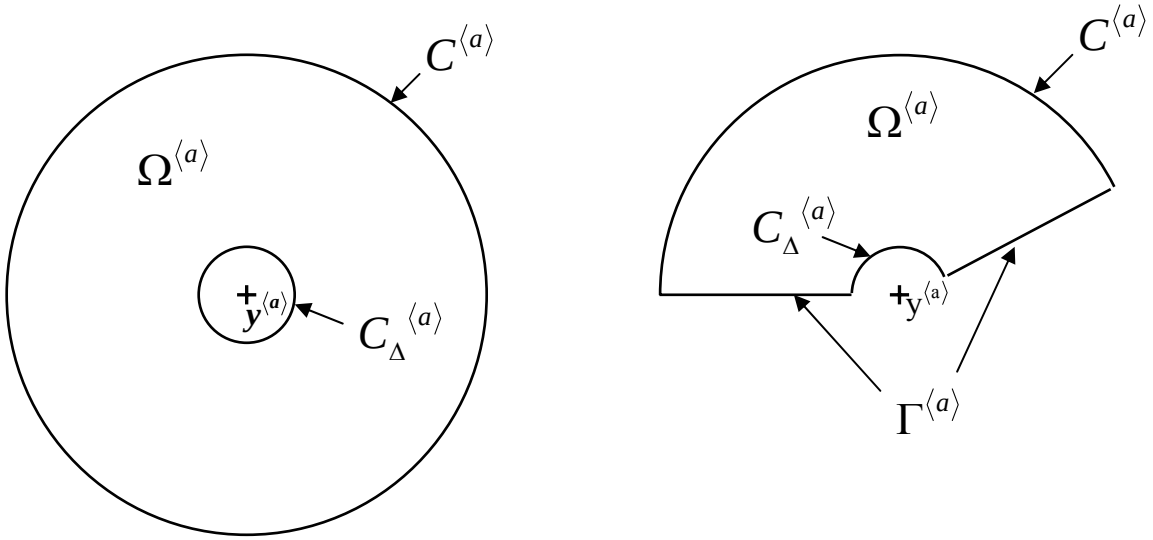


Figure 3.2 Exclusion of a tiny sphere Ω_Δ of radius Δ centered a node for removing the strong singularity



When $\mathbf{y}^{\langle a\rangle}$ is an interior node

$$\partial\Omega^{\langle a\rangle} = C^{\langle a\rangle} + C_\Delta{}^{\langle a\rangle} + \Gamma^{\langle a\rangle}$$

with $\Gamma^{\langle a\rangle} = \varnothing$

When $\mathbf{y}^{\langle a\rangle}$ is a boundary node

$$\partial\Omega^{\langle a\rangle} = C^{\langle a\rangle} + C_\Delta{}^{\langle a\rangle} + \Gamma^{\langle a\rangle}$$

Figure 3.3 Schematic diagram showing the internal boundary angle $\theta = \theta_2 - \theta_1$ at node $\mathbf{y}^{\langle a \rangle}$ on the boundary

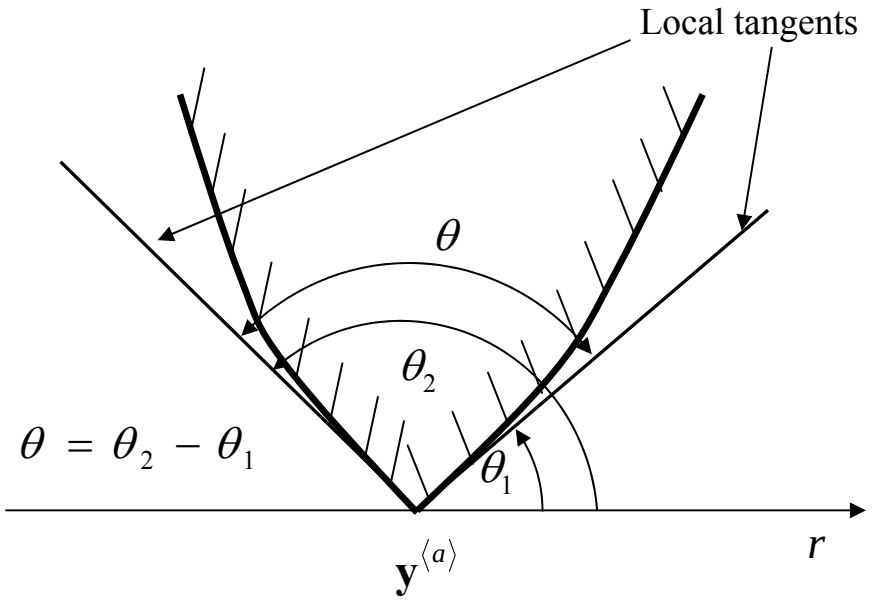


Figure 3.4 Schematic diagram illustrating the meaning of local sub-domain and support domain for node $\mathbf{y}^{\langle a \rangle}$ and node $\mathbf{y}^{<b>}$

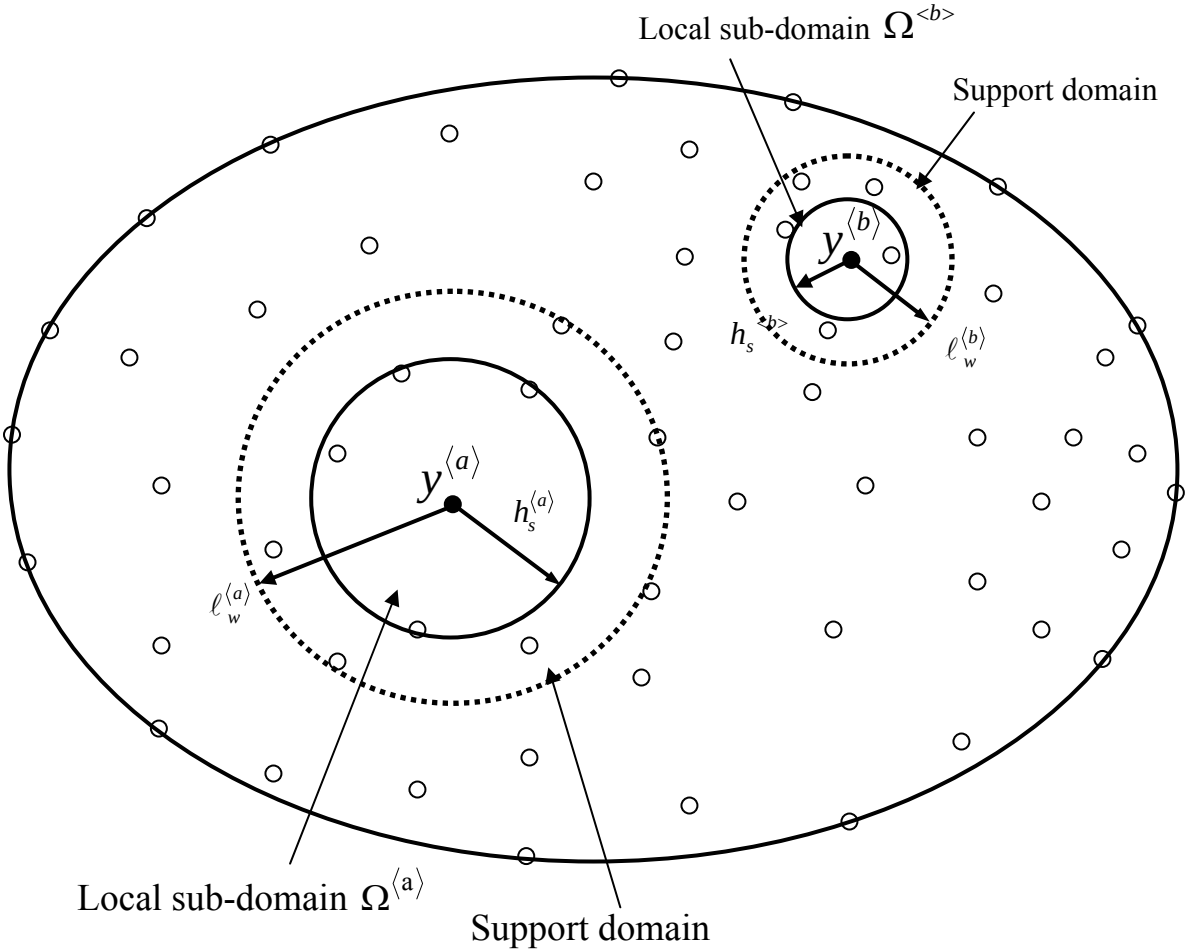


Figure 3.5 Schematic diagram showing the domain of influence for node $\mathbf{y}^{\langle a \rangle}$ and the domain of definition for a point $\mathbf{x}$

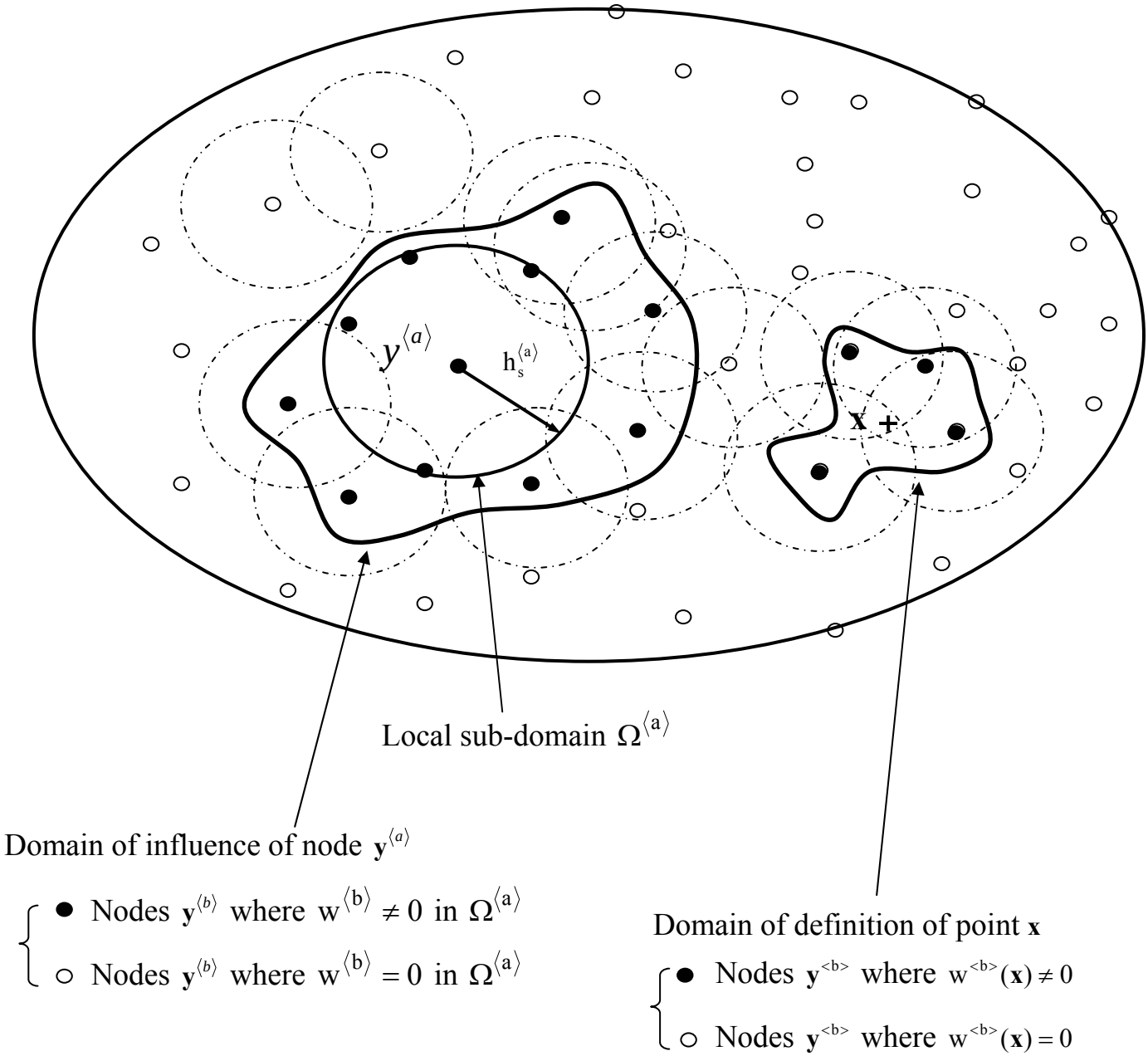
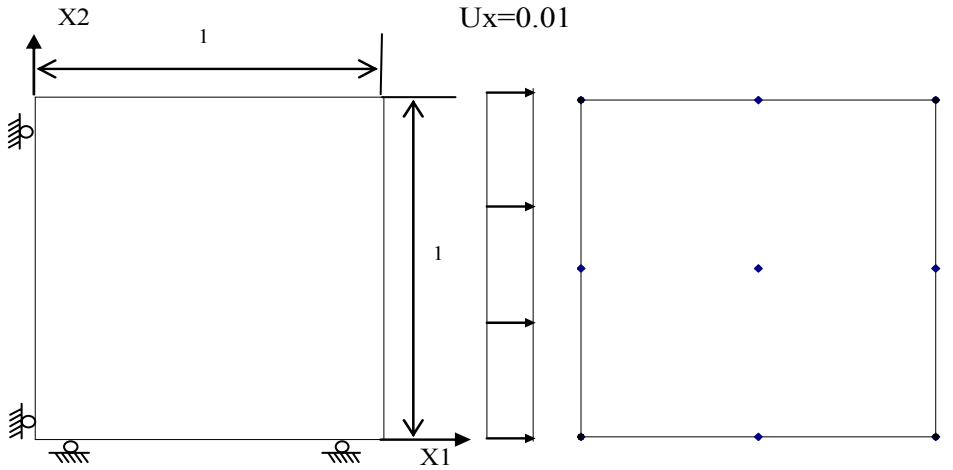
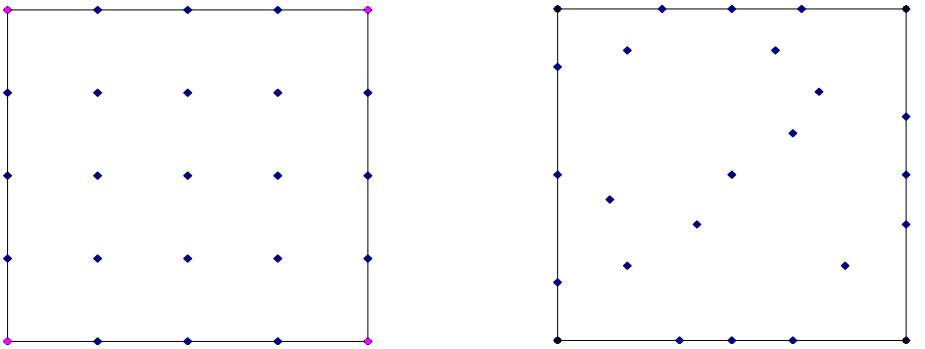


Figure 3.6 A square plate for the patch test with three meshless models



(a) A square plate (b) 9 regular nodes



(d) 25 regular nodes (d) 25 irregular nodes

Figure 3.7 Deformed meshless model using spline weight function and linear basis with 9 nodes in plane strain condition

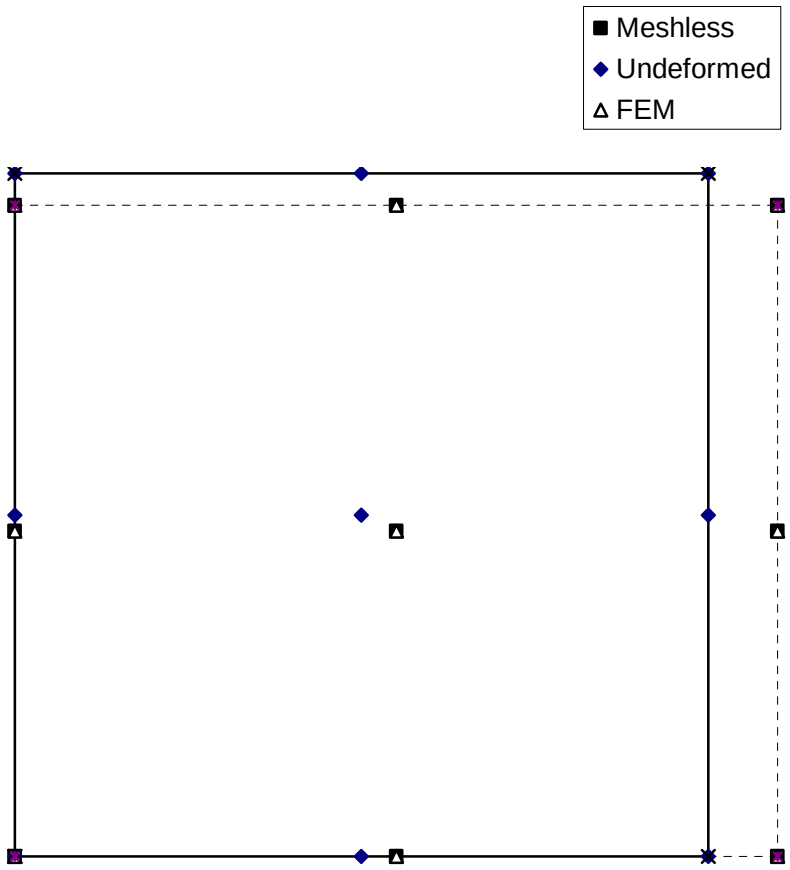


Figure 3.8 Deformed meshless model using spline weight function and linear basis with 9 nodes in plane stress condition

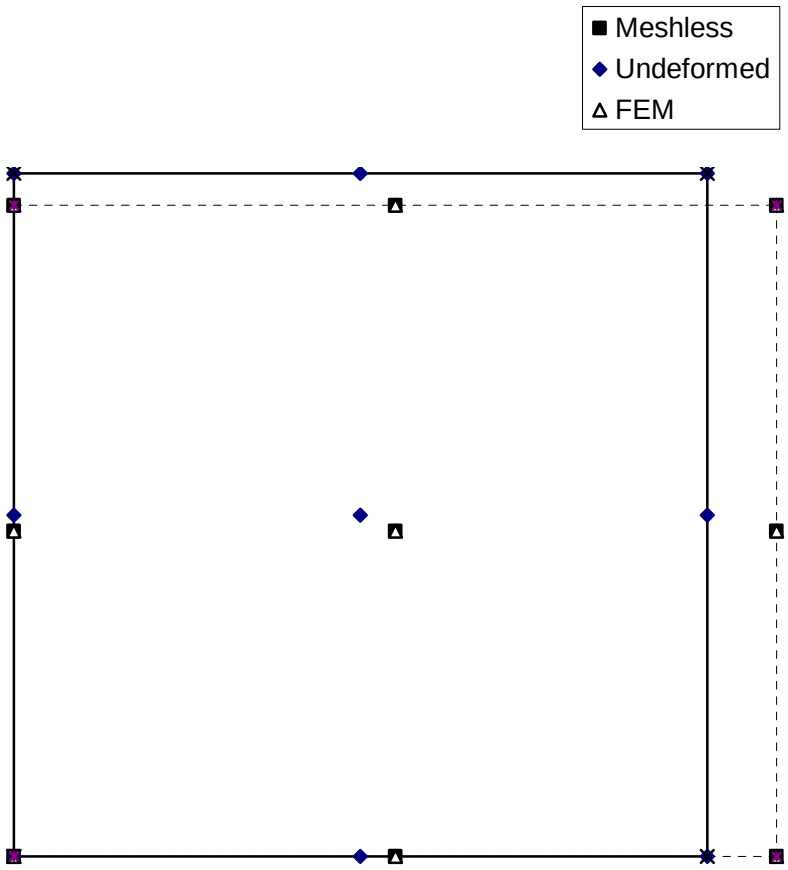
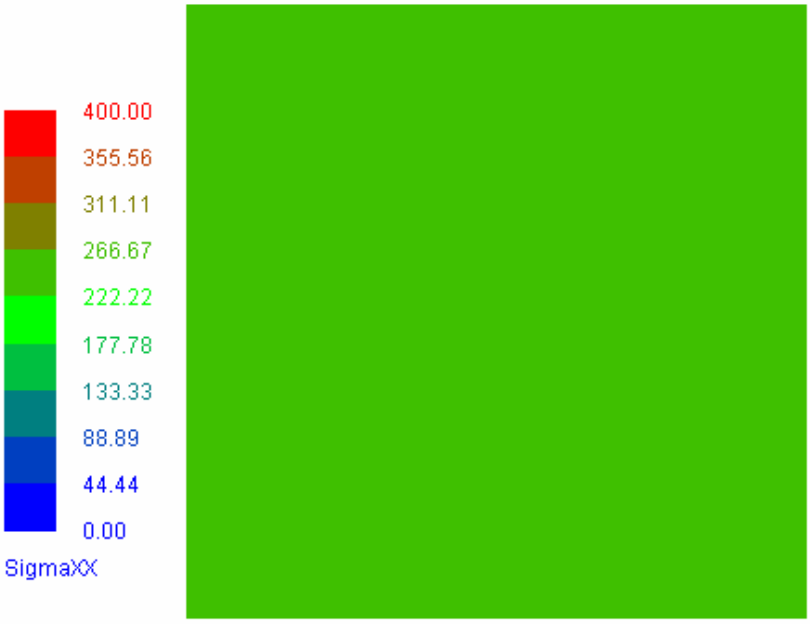
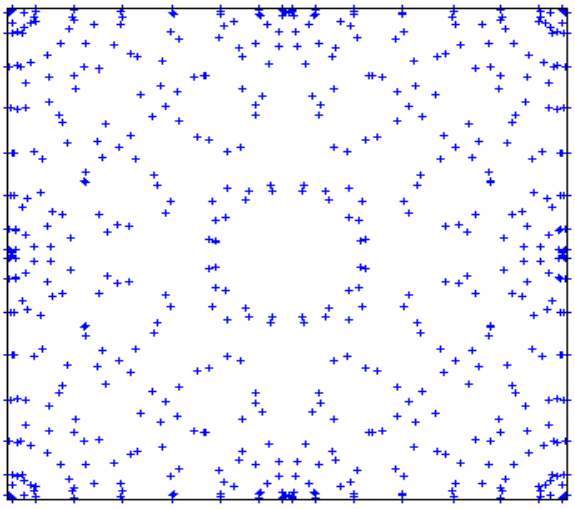


Figure 3.9 (a) Distribution of σ_{11} for 9 node model in plane stress condition. (b) The status of Gauss integration points (plus sign means plastic state)



(a)



(b)

Figure 3.10 The change of σ_{11} along with load increase

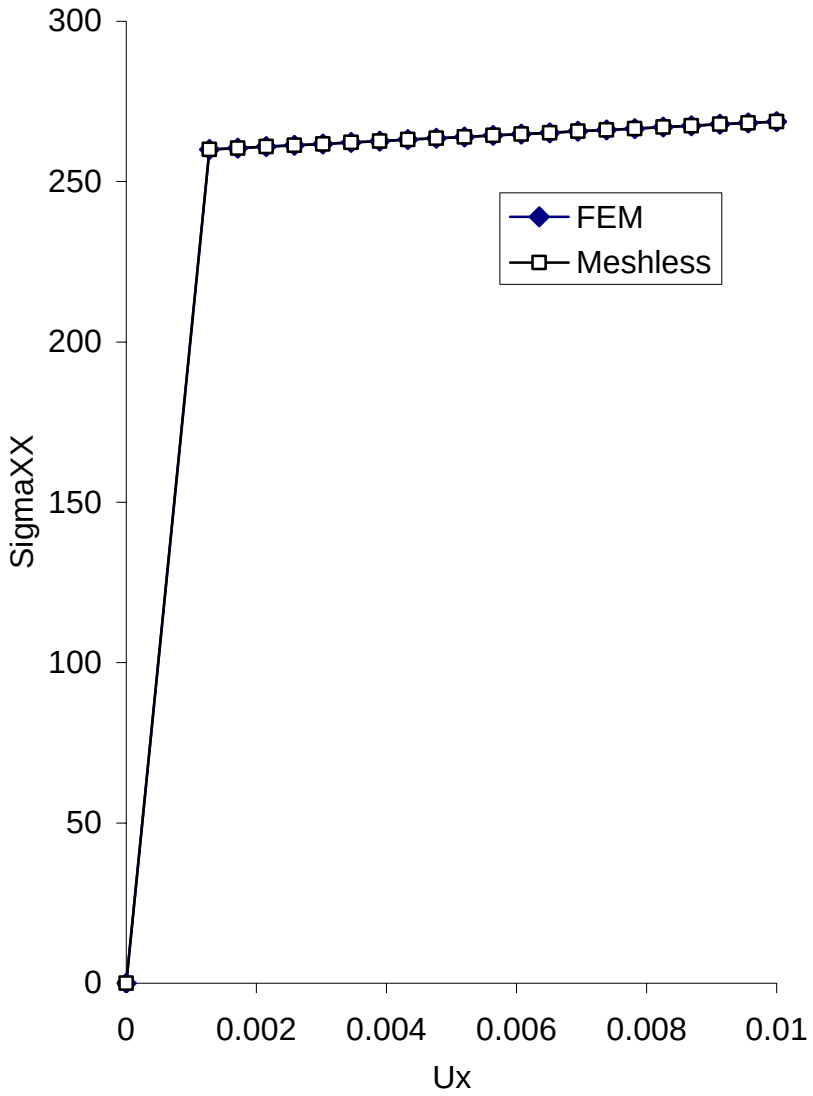
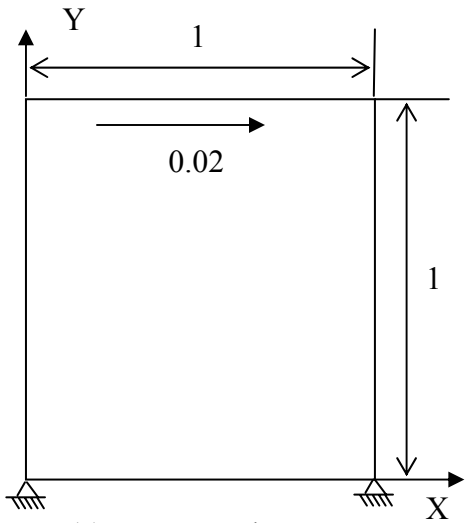
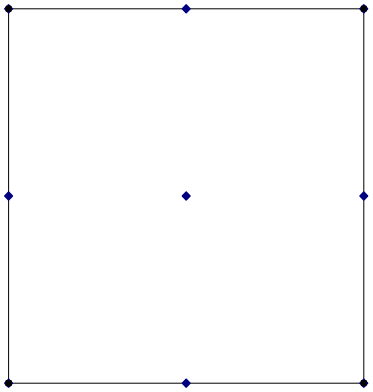


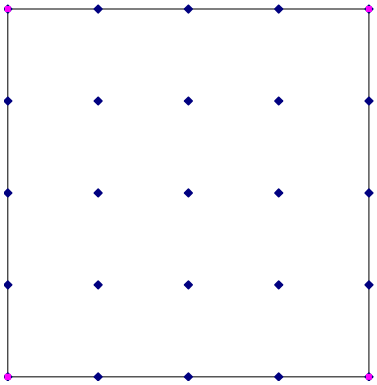
Figure 3.11 (a) A square plate for the shear tests. (b) Meshless model with 9 regular nodes. (c) Meshless model with 25 regular nodes. (d) Meshless model with 25 irregular nodes



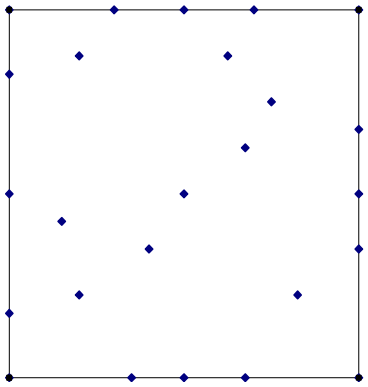
(a) A square plate



(b) 9 regular nodes



(c) 25 regular nodes



(d) 25 irregular nodes

Figure 3.12 Deformed meshless model using spline weight function and linear basis with 25 irregular nodes in plane strain condition

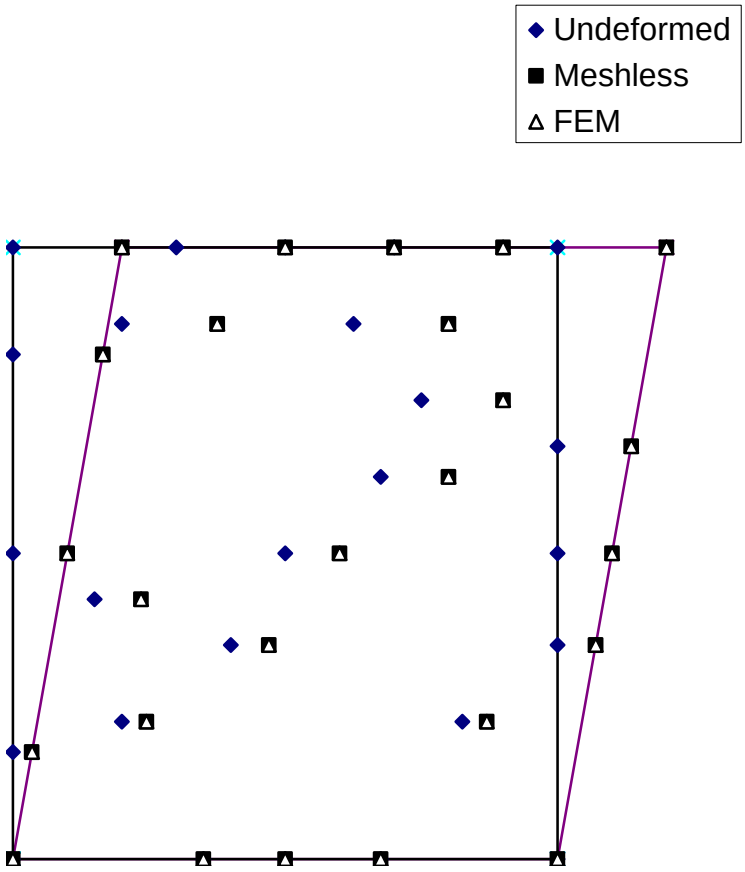


Figure 3.13 Deformed meshless model using spline weight function and linear basis with 25 irregular nodes in plane stress condition

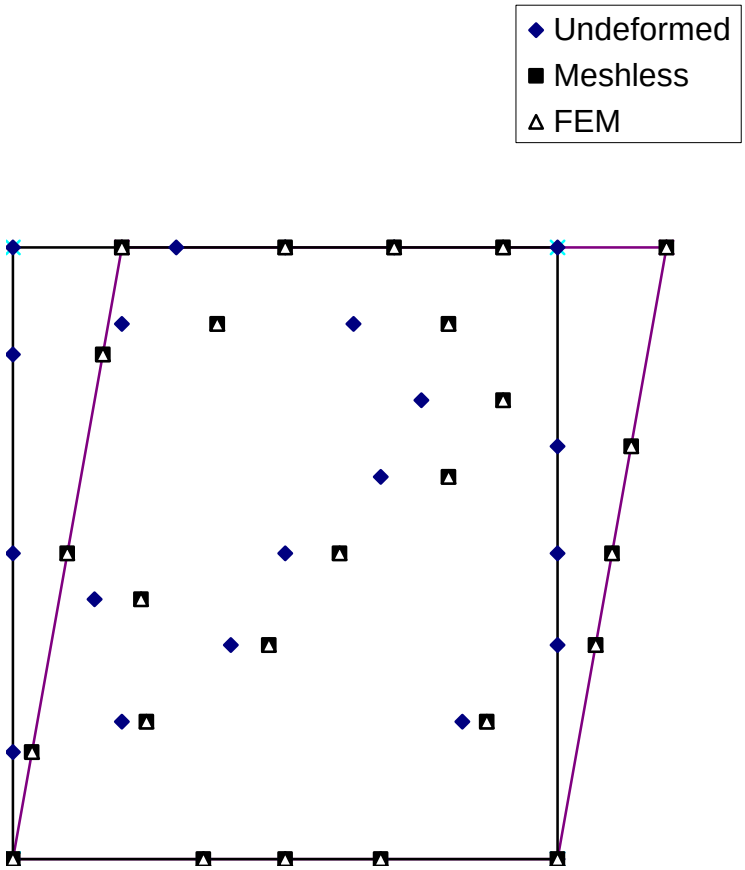
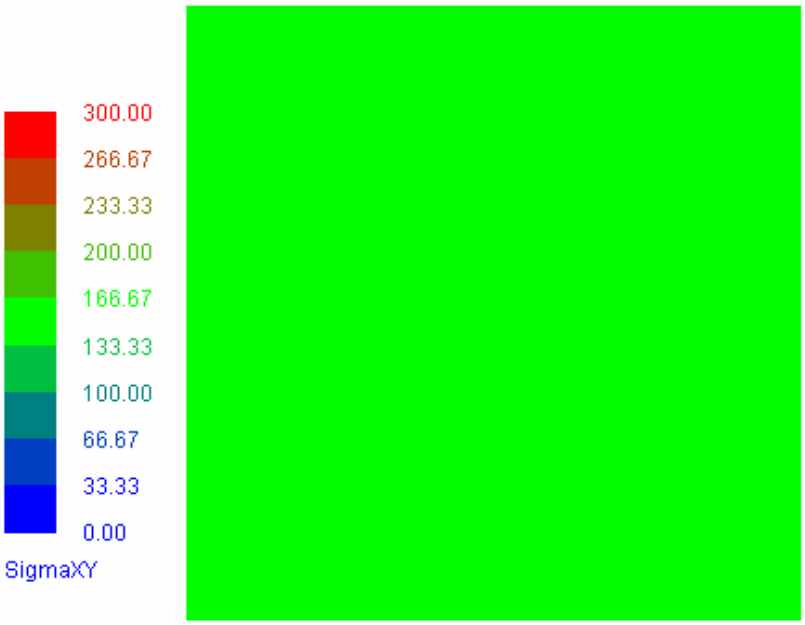
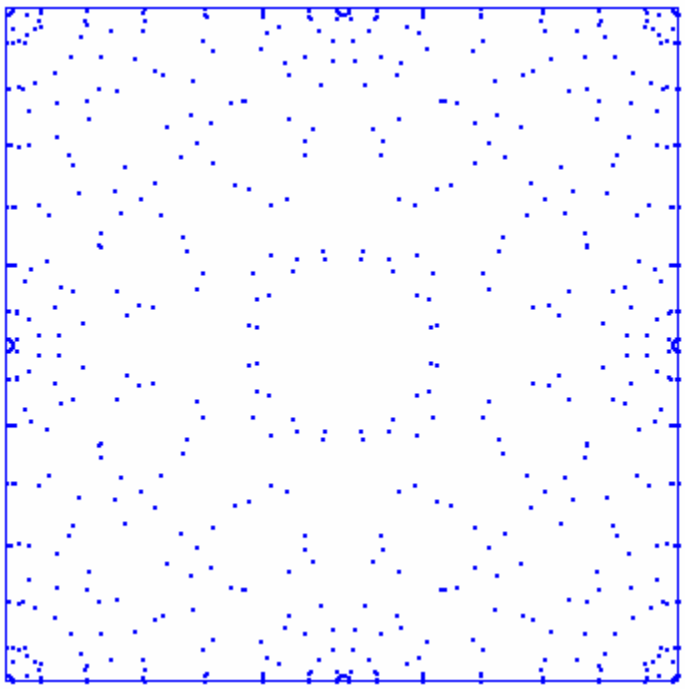


Figure 3.14 (a) Distribution of σ_{12} for 9 node model in plane stress condition. (b) The status of Gauss integration points (plus sign means plastic state)



(a)



(b)

Figure 3.15 The change of σ_{12} along with the increase of shear strain γ

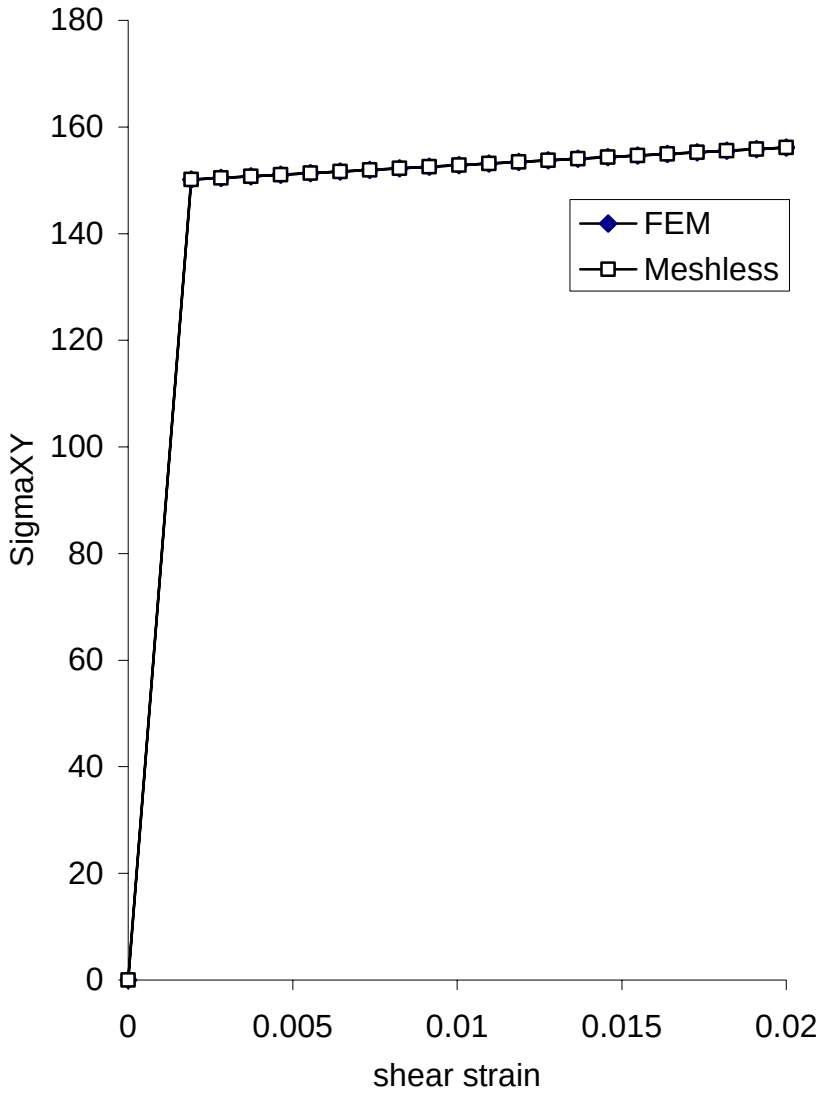


Figure 3.16 The upper right quadrant of the plate with a hole

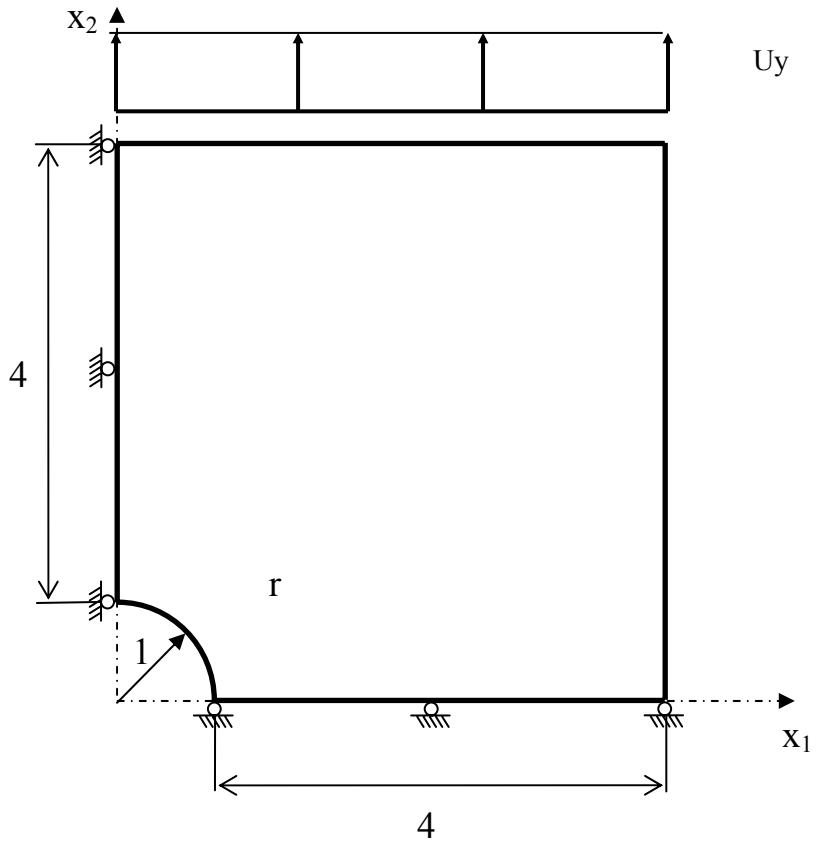


Figure 3.17 The deformed meshless model with Uy=0.01 m

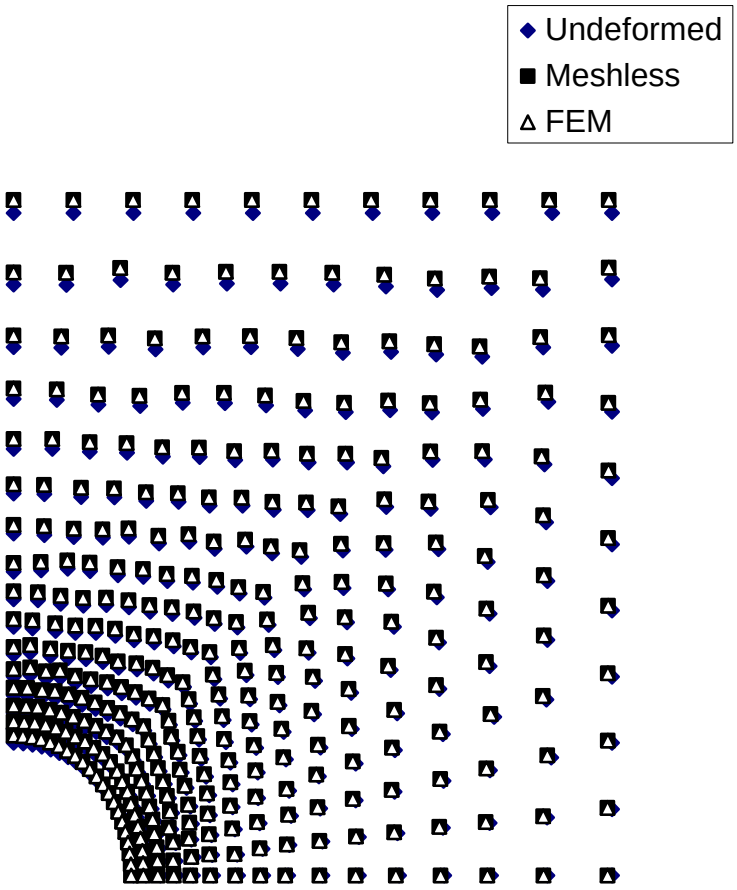


Figure 3.18 Distribution of σ_{22} along x2=0 of the 336 node model with Uy=0.01 m.

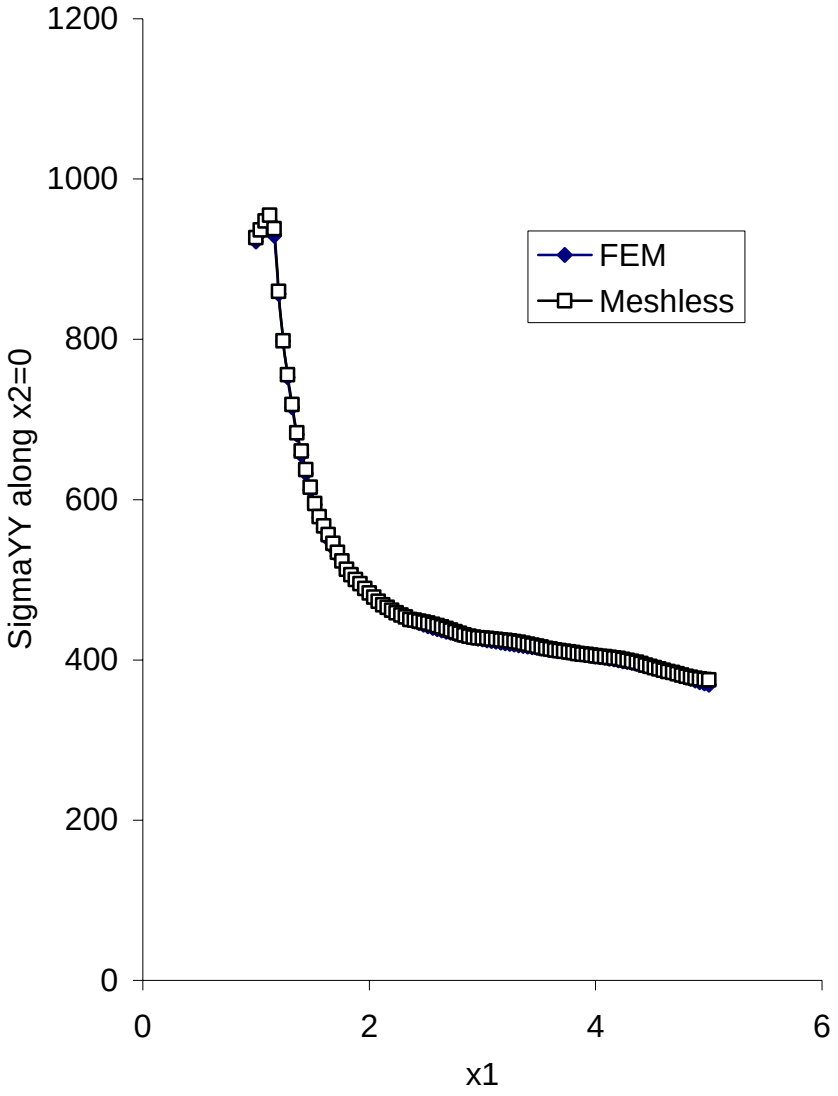


Figure 3.19 Distribution of von Mises stress along x2=0 of the 336 node model with Uy=0.01 m.

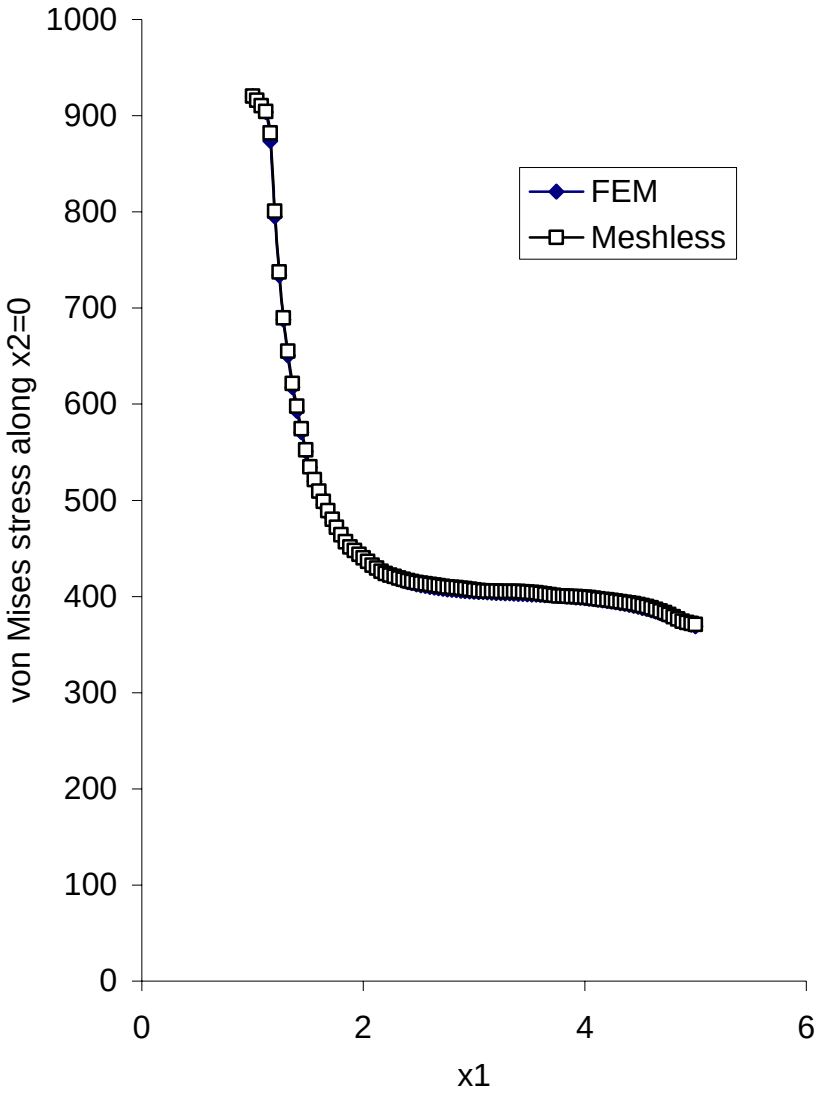


Figure 3.20 The deformed meshless model with Uy=0.014 m

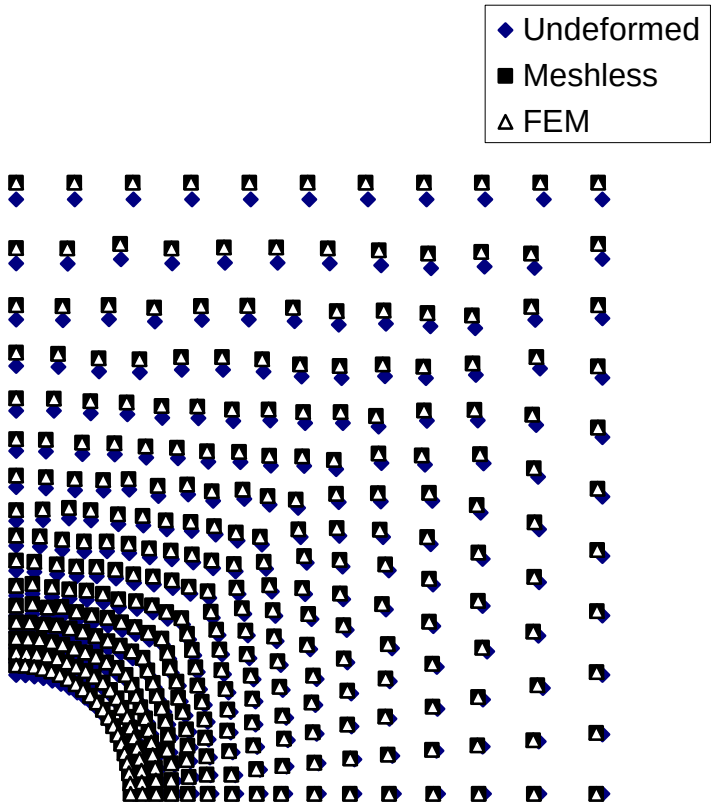


Figure 3.21 Distribution of σ_{22} along x2=0 of the 336 node model with Uy=0.014 m.

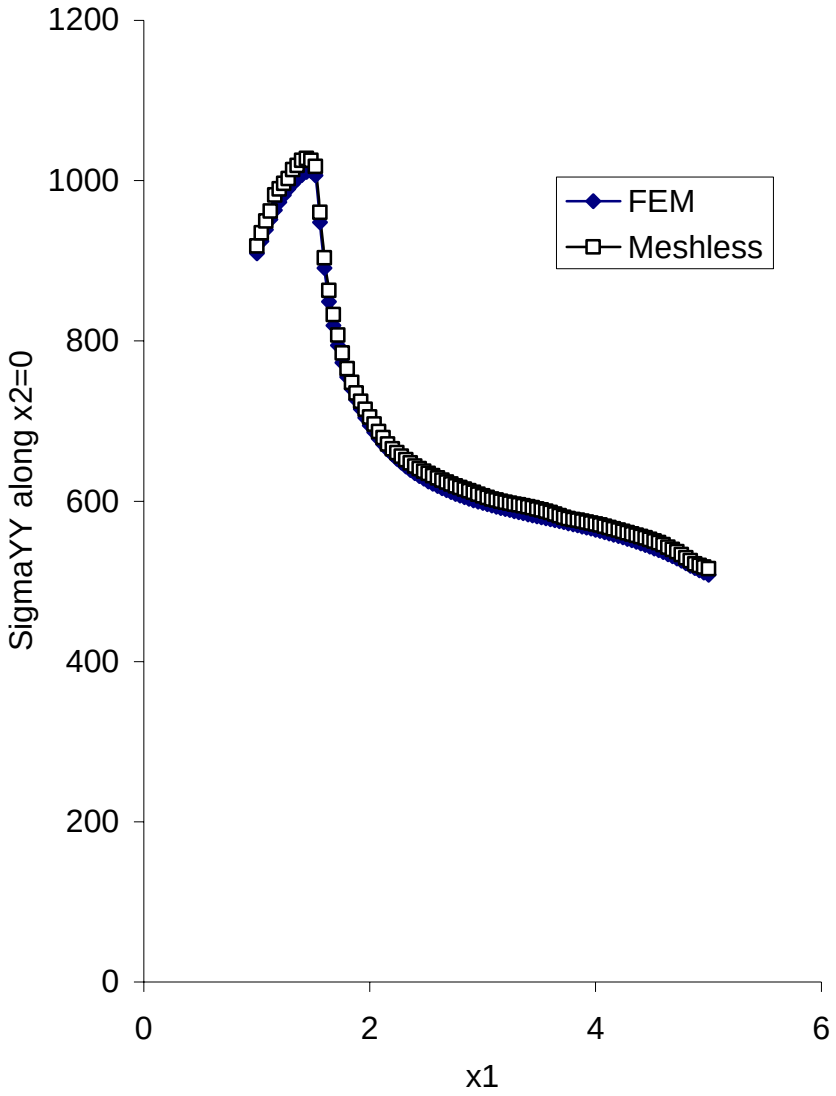


Figure 3.22 Distribution of von Mises stress along x2=0 of the 336 node model with Uy=0.014 m.

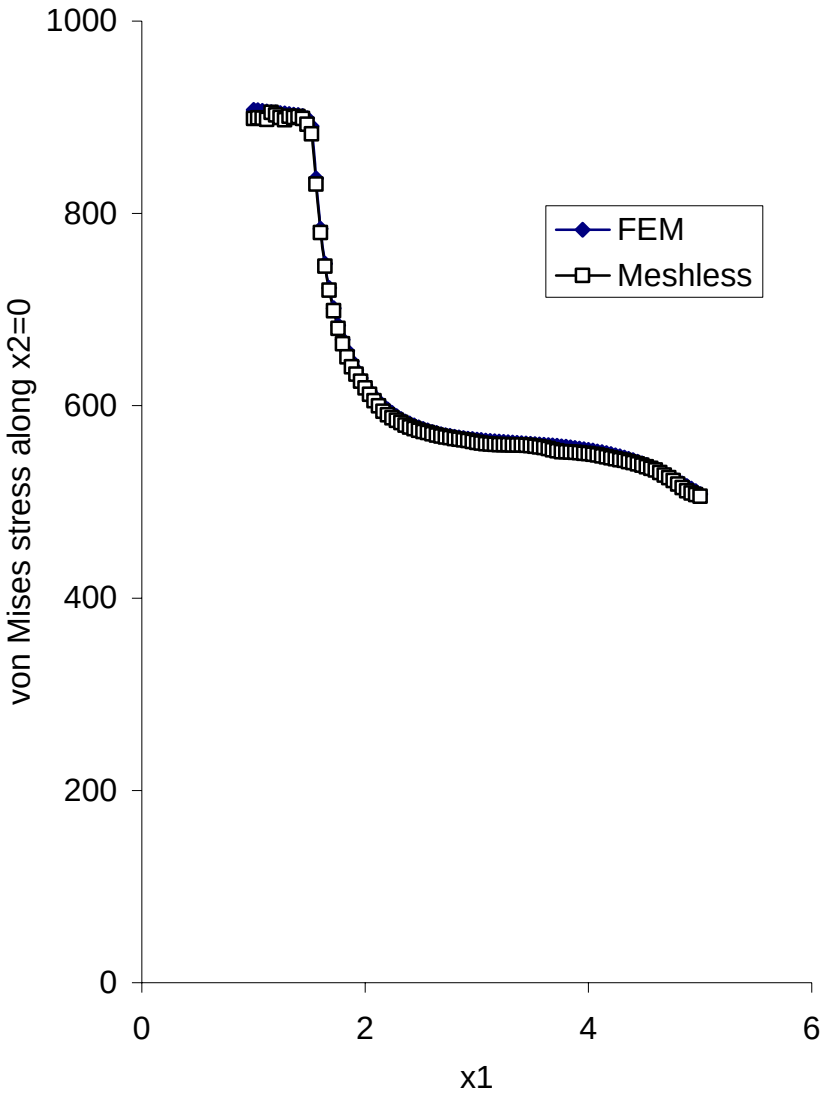


Figure 3.23 The deformed meshless model with Uy=0.018 m

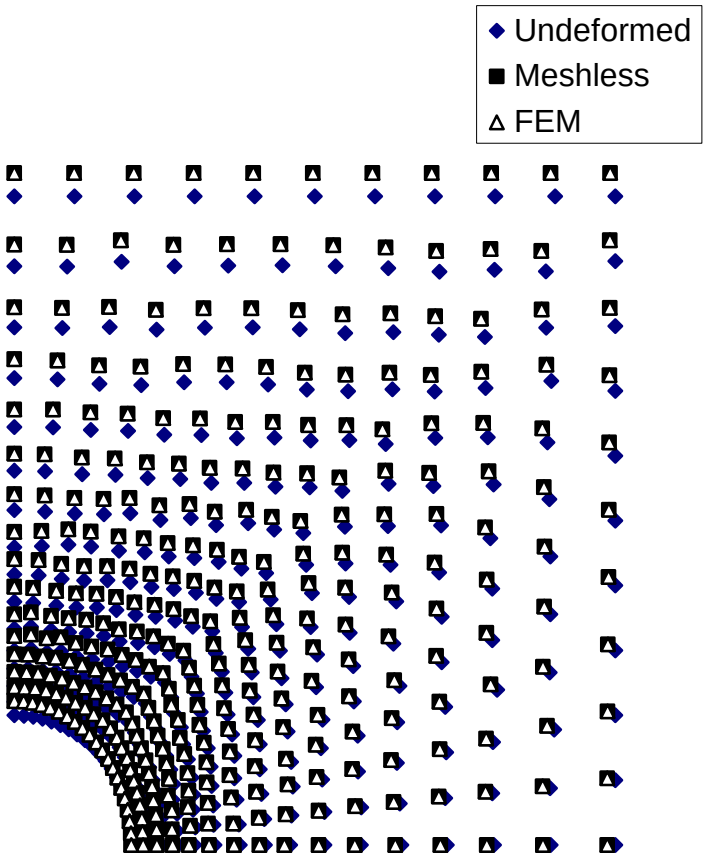


Figure 3.24 Distribution of σ_{22} along x2=0 of the 336 node model with Uy=0.018 m.

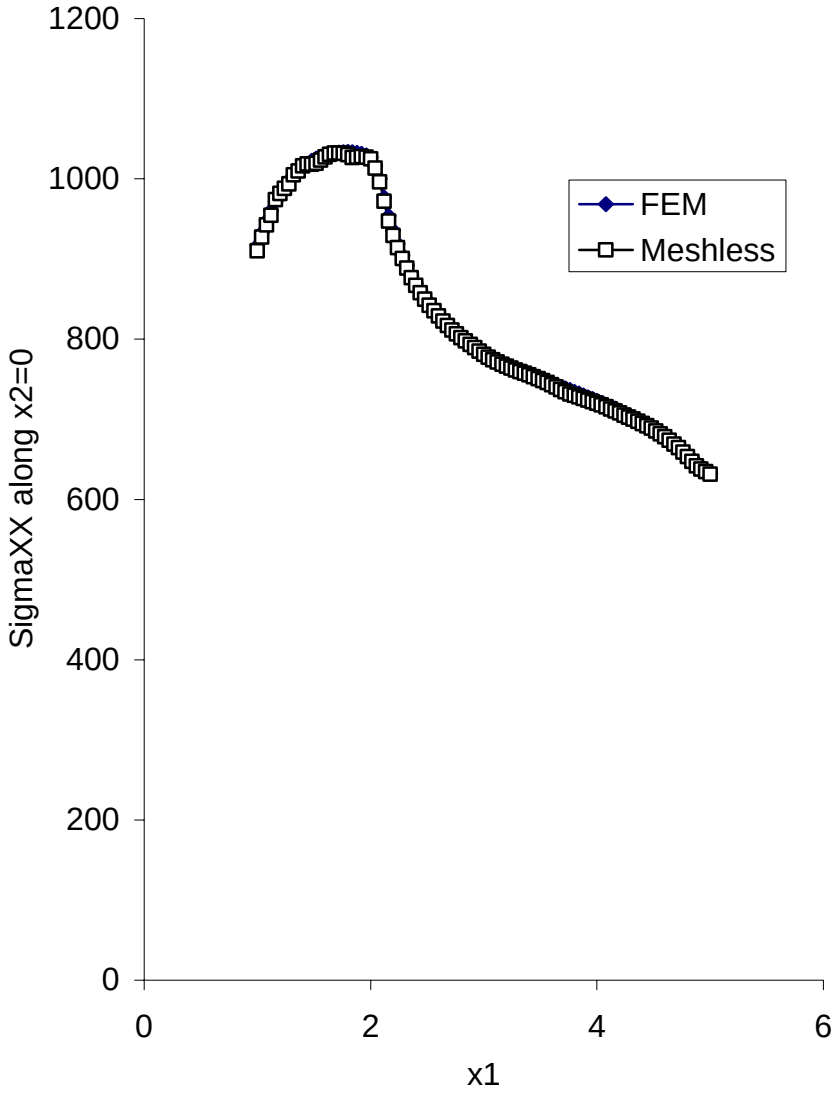


Figure 3.25 Distribution of von Mises stress along x2=0 of the 336 node model with Uy=0.018 m

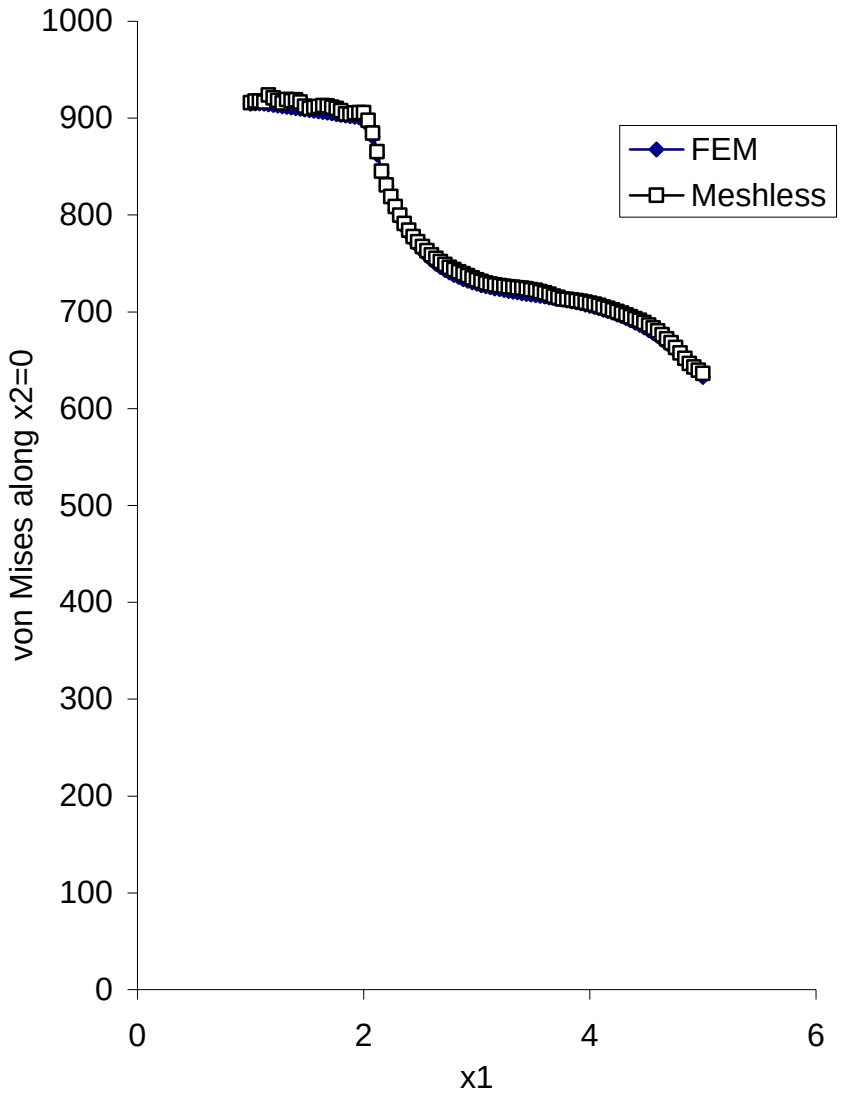
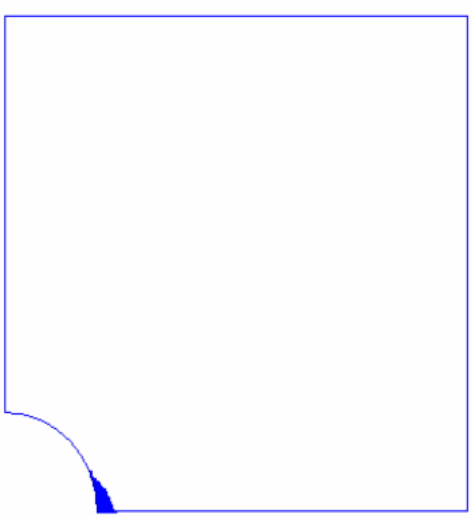
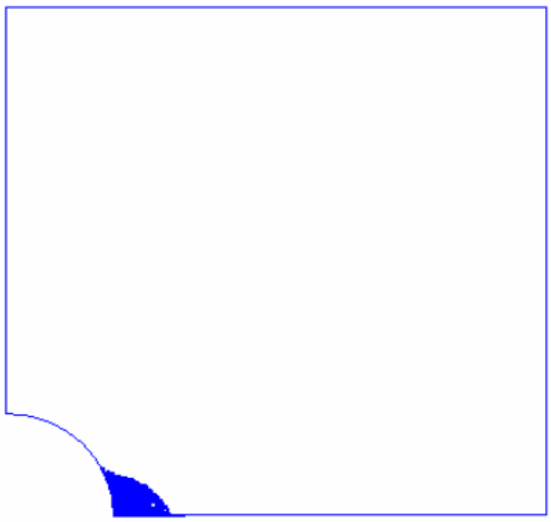


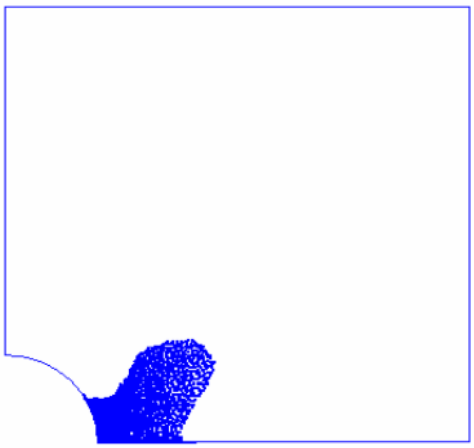
Figure 3.26 The spreading of the plastic zone for hole problem



(a) $\frac{Uy}{b} = 1.303$



(b) $\frac{Uy}{b} = 1.824$



(c) $\frac{Uy}{b} = 2.346$

Figure 3.27 (a) Thick-walled cylinder subjected to a gradually increasing internal pressure. (b) the upper right quadrant of the plate was modeled

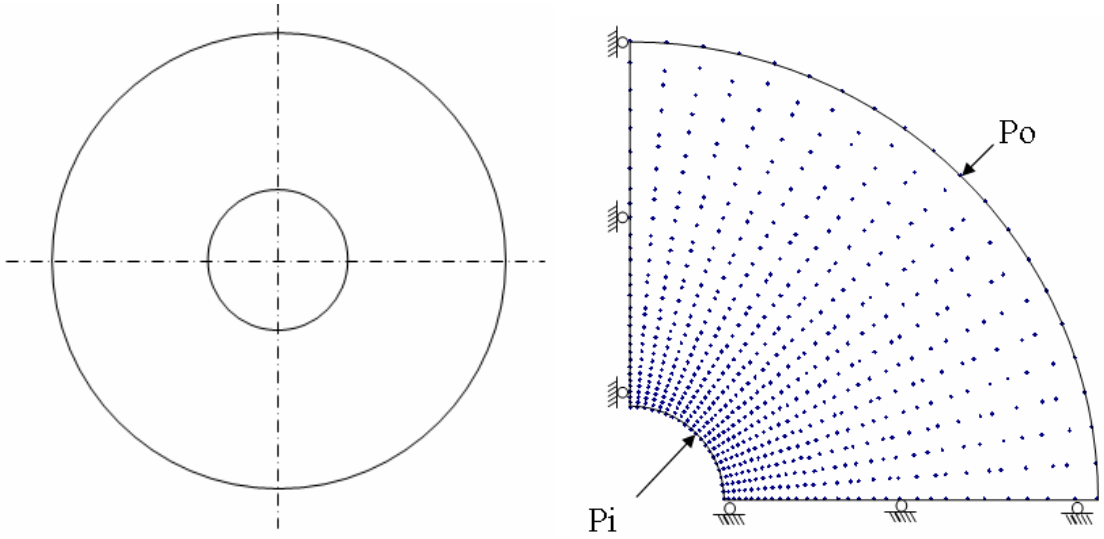


Figure 3.28 The effect of the sub-domain radius on the numerical results for linear elasticity

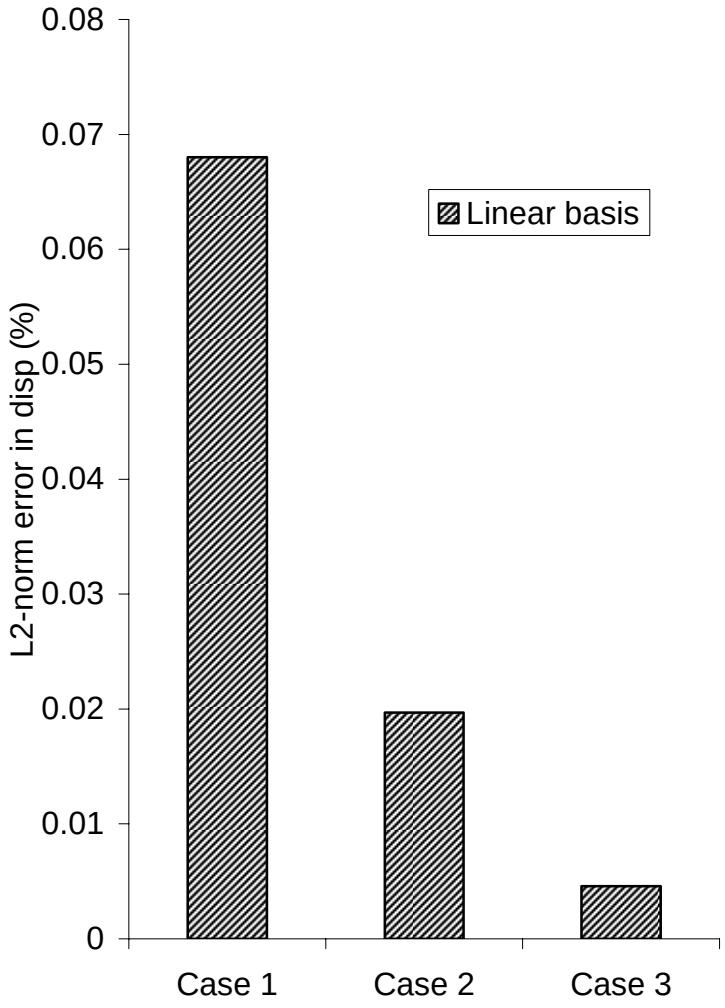


Figure 3.29 The undeformed model (diamond), deformed meshless model using case 3 to set the sub-domain radius for elastic case

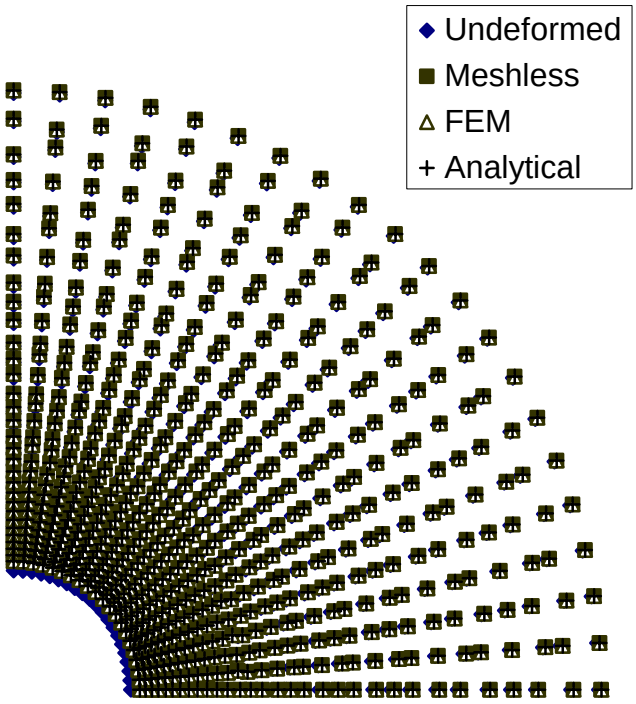


Figure 3.30 The undeformed model (diamond), deformed meshless model using case 3 to set the sub-domain radius for elastoplastic case with internal pressure 261 MPA

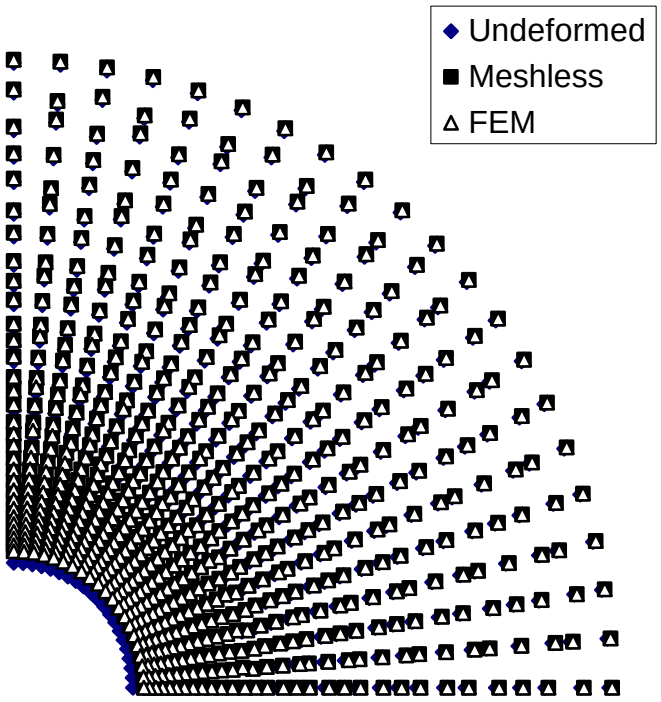
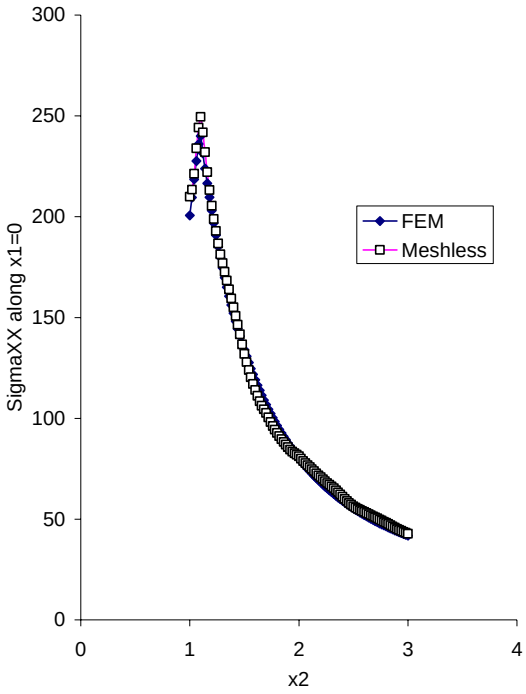
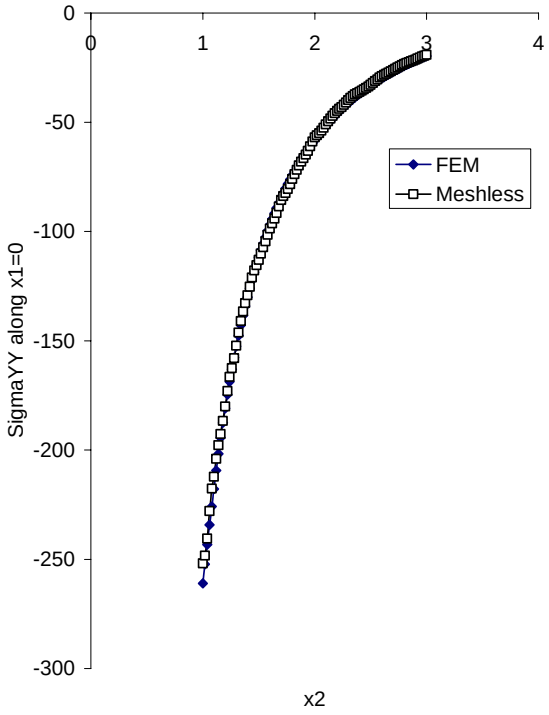


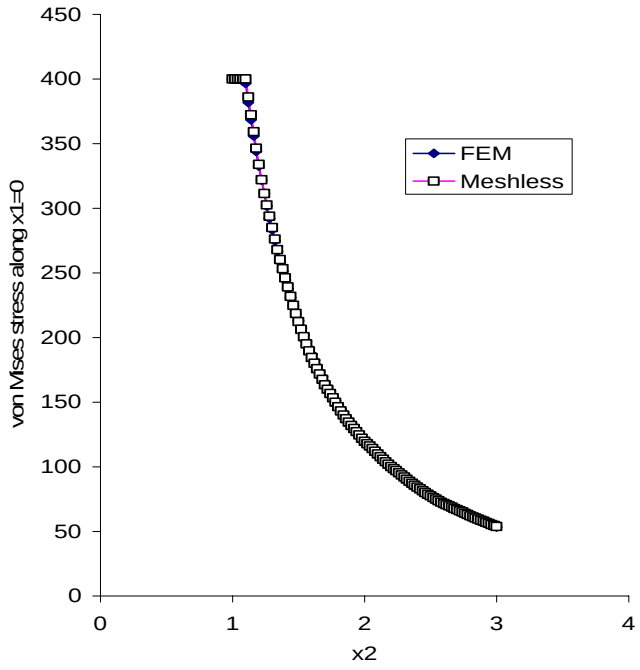
Figure 3.31 The distribution of σ_{11} , σ_{22}, and von Mises stress along x1=0 of the 651 node model with internal pressure 261 MPA. (a) σ_{11}; (b) σ_{22}; (c) von Mises Stress



(a)



(b)



(c)

Figure 3.32 The undeformed model (diamond), deformed meshless model with internal pressure 319 MPA for elastoplastic case

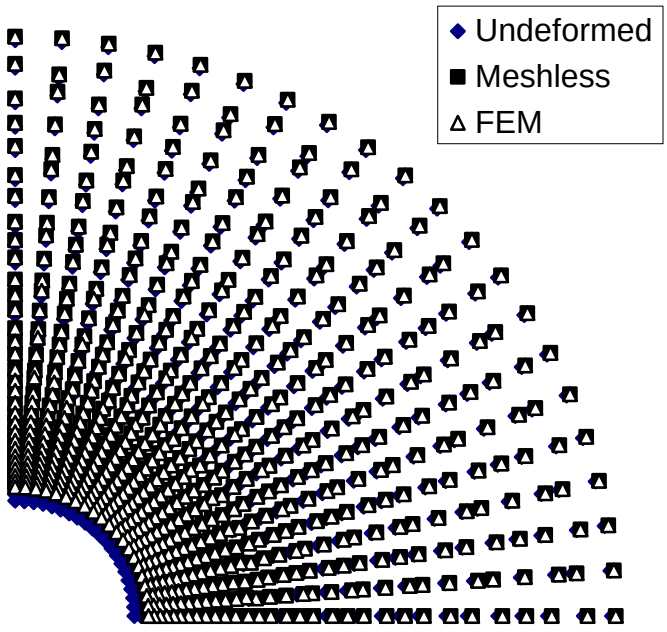
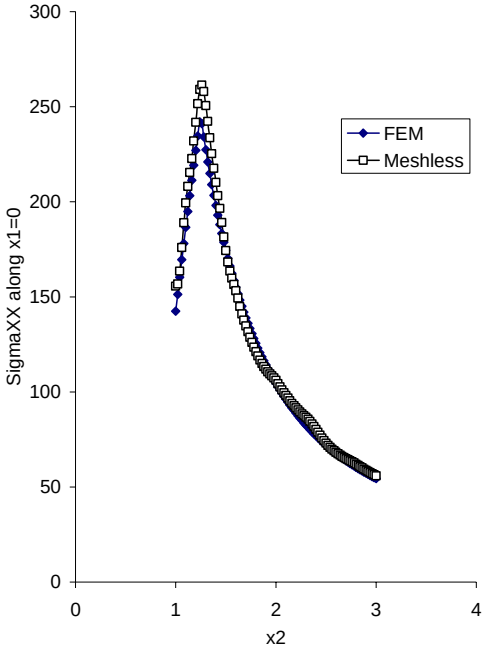
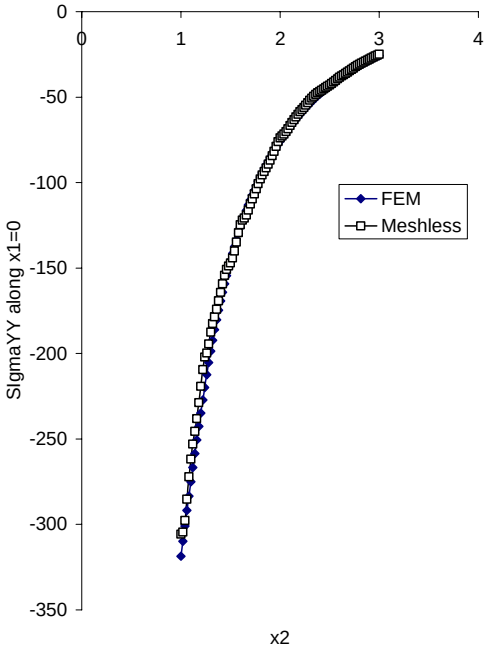


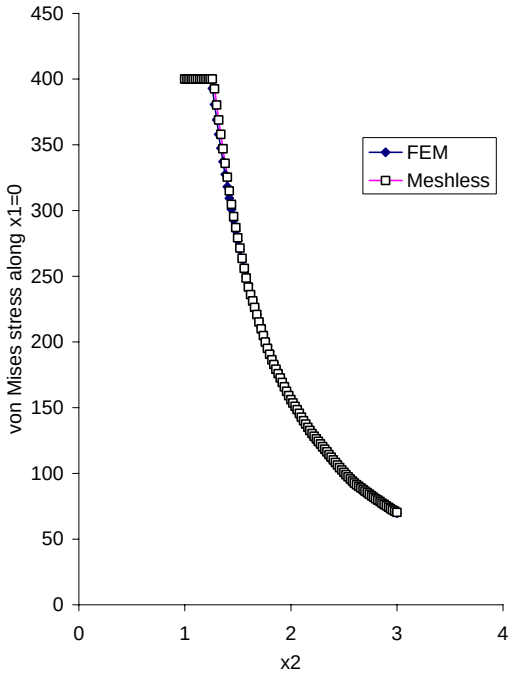
Figure 3.33 The distribution of σ_{11} , σ_{22} , and von Mises stress along x1=0 of the 651 node model with internal pressure 319 MPA. (a) σ_{11}; (b) σ_{22}, (c) von Mises Stress



(a)



(b)



(c)

Figure 3.34 The undeformed model (diamond), deformed meshless model with internal pressure 345 MPA for elastoplastic case

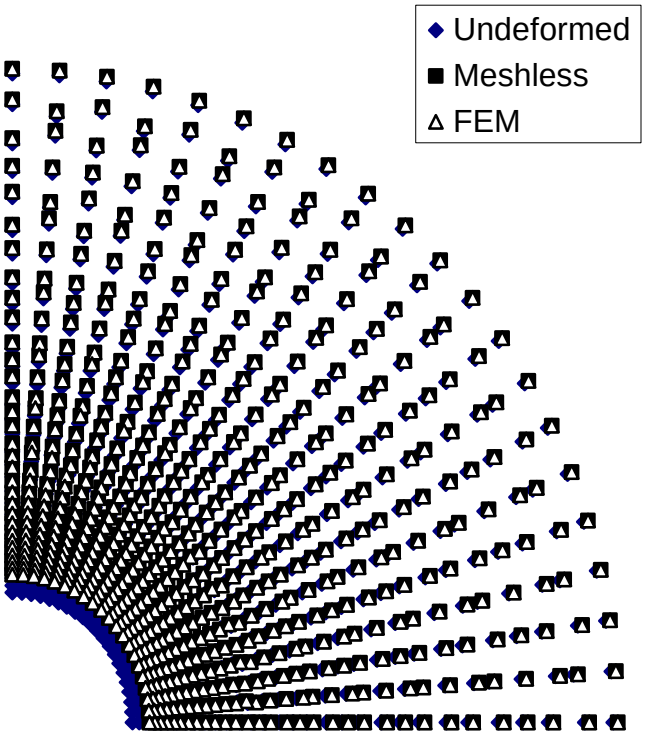
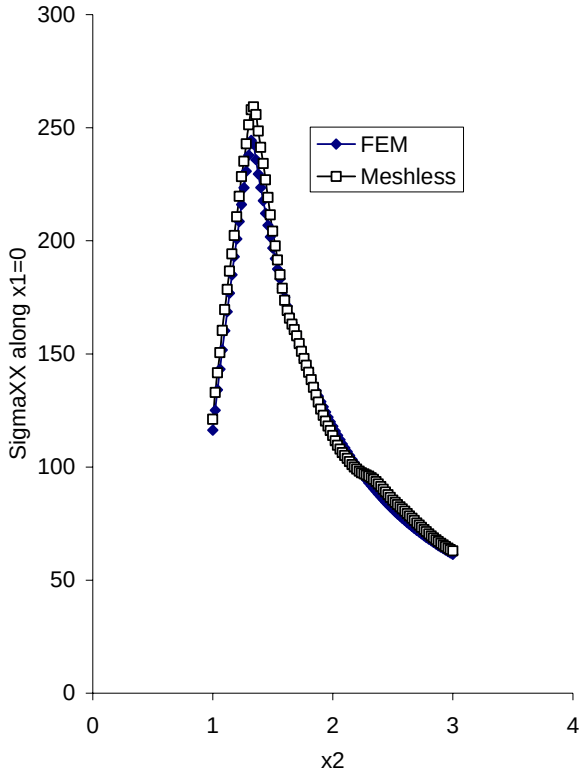
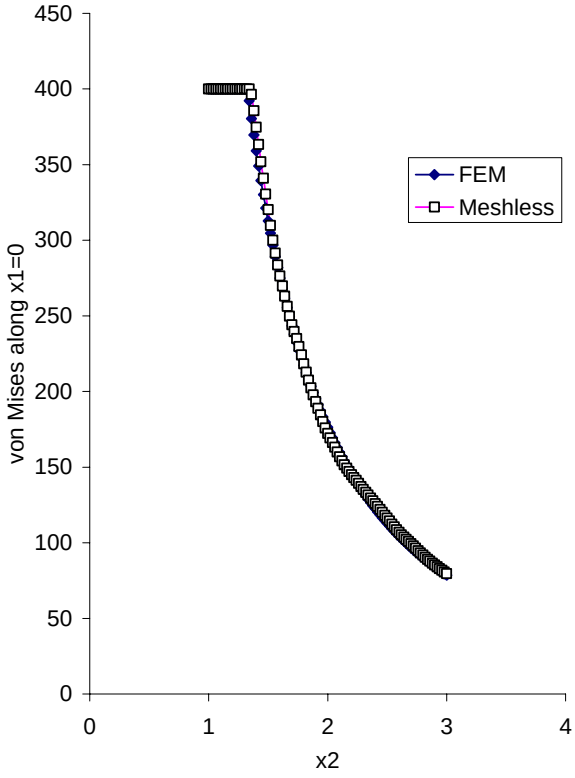


Figure 3.35 The distribution of σ_{11} and von Mises stress along x1=0 of the 651 node model with internal pressure 345 MPA. (a) σ_{11}; (b) von Mises Stress



(a)



(b)

Figure 3.36 The undeformed model (diamond), deformed meshless model with internal pressure 445 MPA for elastoplastic case

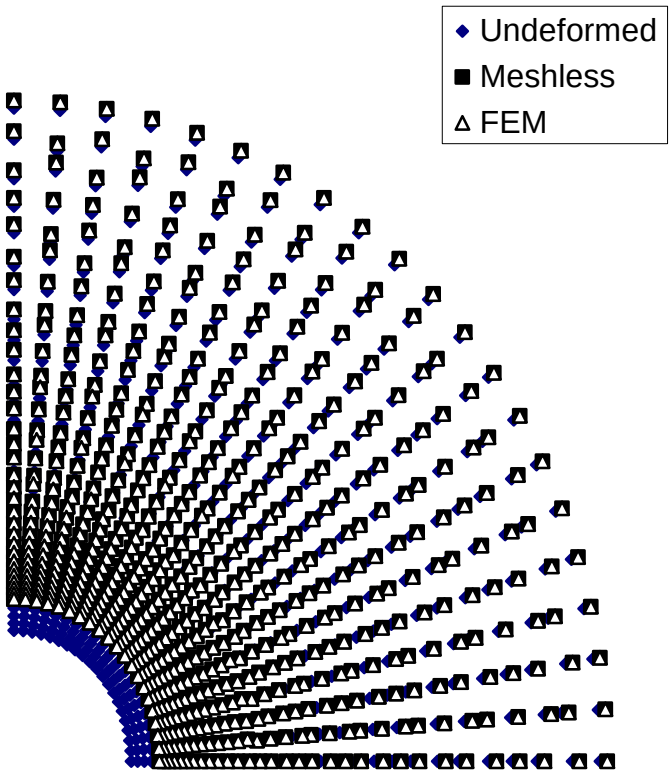
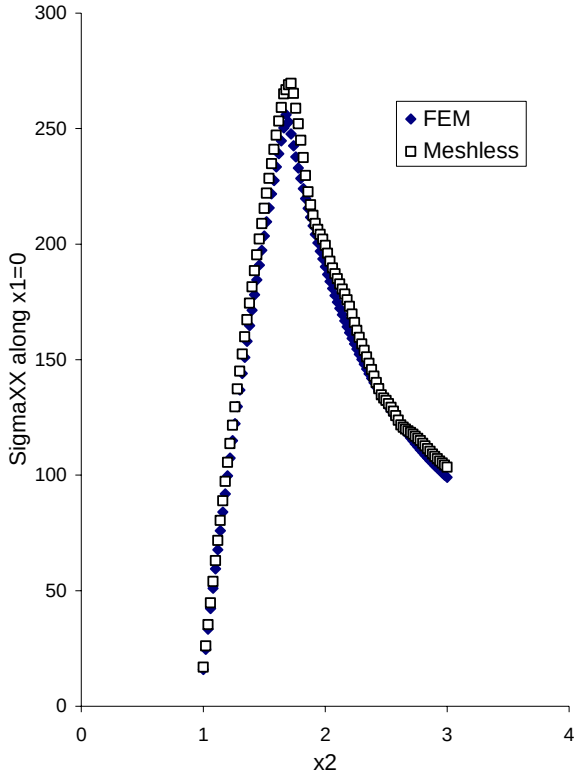
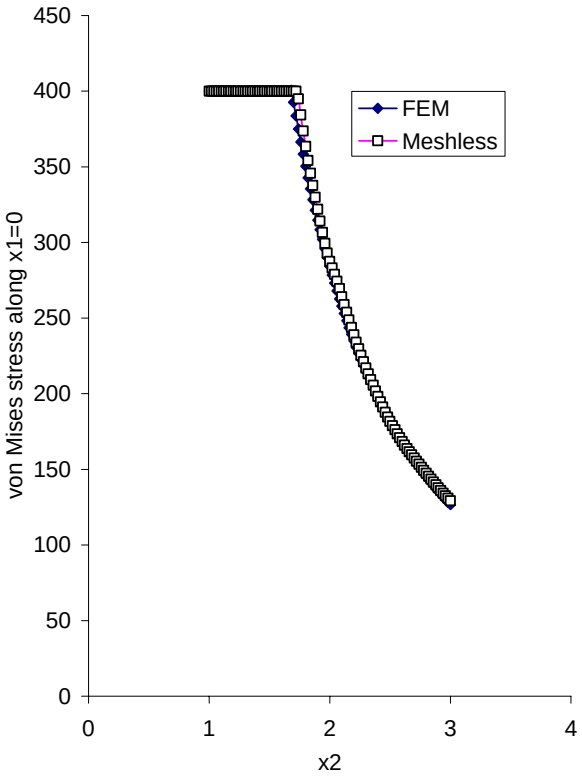


Figure 3.37 The distribution of σ_{11} and von Mises stress along x1=0 of the 651 node model with internal pressure 445 MPA. (a) σ_{11}; (b) von Mises Stress

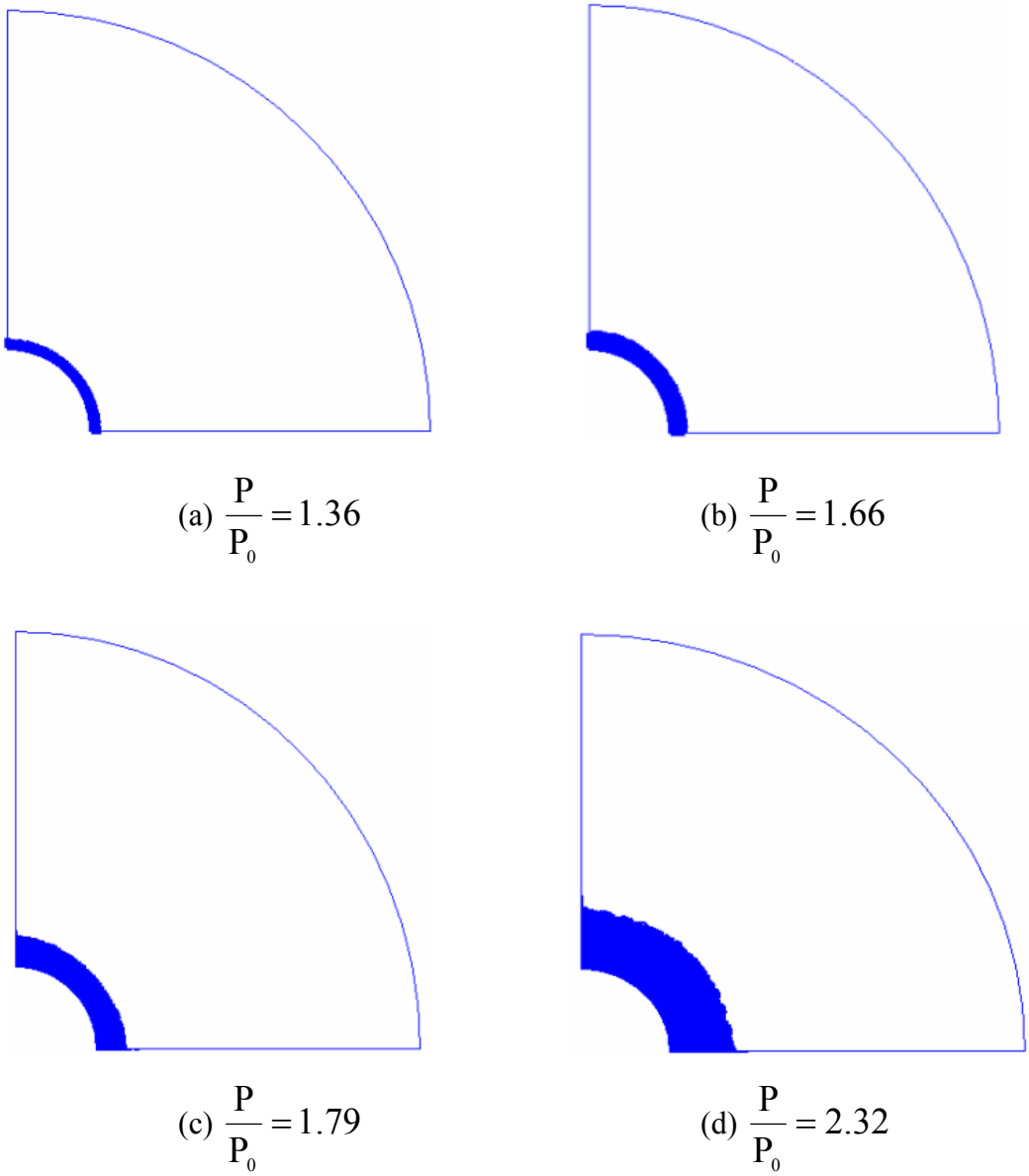


(a)



(b)

Figure 3.38 The spreading of the plastic zone for cylinder



(a) $\frac{P}{P_0} = 1.36$ (b) $\frac{P}{P_0} = 1.66$

(c) $\frac{P}{P_0} = 1.79$ (d) $\frac{P}{P_0} = 2.32$

Table 3.1 Special numerical integration for functions containing logarithmic singularity (8 integration points)

$$\int_0^1 f(r)\ln\left(\frac{1}{r}\right)dr = \sum_{i=1}^{N} w_i f(r_i)$$

Abscissas (r_i)	Weights (w_i)
0.0133 2024 4160 8925	0.1644 1660 4728 0030
0.0797 5042 9013 8949	0.2375 2561 0023 3060
0.1978 7102 9326 1880	0.2268 4198 4431 9190
0.3541 5399 4351 9090	0.1757 5407 9006 0700
0.5294 5857 5234 9170	0.1129 2403 0246 7590
0.7018 1452 9939 1000	0.0578 7221 0717 7821
0.8493 7932 0441 1070	0.0209 7907 3742 1330
0.9533 2645 0056 3600	0.0036 8640 7104 0276

Table 3.2 L_2-norm errors for patch test under monotonic loading

	9 nodes	25 nodes-regular	25 nodes-irregular
Plane strain	0.0024	0.0027	0.0030
Plane stress	2.6E-5	2.4E-5	3.3E-5

Table 3.3 L_2-norm errors for patch test under cycle loading

	9 nodes	25 nodes-regular	25 nodes-irregular
Plane strain	0.0027	0.0021	0.0038
Plane stress	3.7E-5	4.3E-5	2.6E-5

Table 3.4 L_2**-norm errors for shear tests**

Case 1	9 nodes	25 nodes-regular	25 nodes-irregular
Plane strain	1.9E-16	4.3E-16	7.6E-15
Plane stress	3.7E-11	6.1E-10	9.7E-10

CHAPTER 4 - Meshless Integral Method for Elastoplastic Materials with Large Deformation

Manuscript to be submitted to:

International Journal for Numerical Methods in Engineering – 2007

Authors' Names:

Jianfeng Ma, X.J. Xin, and Prakash Krishnaswami

Authors' Affiliations

Department of Mechanical and Nuclear Engineering, Kansas State University, 302 Rathbone Hall, Manhattan, KS 66506-5205

4.1 Abstract

In this paper, the meshless integral method based on the regularized boundary integral equation [1] has been extended to analyze the large deformation of elastoplastic materials. The updated Lagrangian governing integral equation is obtained from the weak form of elastoplasticity over a local sub-domain based on Green-Naghdi's theory, and the moving least-squares approximation is used for meshless function approximation. Green-Naghdi's theory starts with the decomposition of the Green-Lagrange strain into the elastic part and plastic part and considers a J_2 elastoplastic constitutive relation that relates the Green-Lagrange strain to the second Piola-Kirchhoff stress. A simple collocation method is employed to enforce the essential boundary conditions straightforwardly and accurately, and the natural boundary conditions are incorporated in the system governing equation and require no special handling. The solution algorithm for large deformation analysis is discussed in detail. Numerical examples show that this method is accurate and robust.

KEY WORDS: meshless method, large deformation, local boundary integral equation, moving least-squares approximation, subtraction method, singularity removal, elastoplasticity

4.2 Introduction

Large deformation analysis has an important role in solid mechanics. The metal forming processes, such as forging, wire-drawing, rolling and ring test can be treated as small-elastic-large-plastic strain problems. In metal cutting processes, the work piece is subjected to large deformation with elastoplasticity at a high strain rate in the primary deformation which extends from the tip of the cutting tool to the junction between surface of the undeformed work material and the deformed chip. In order to use meshless method to model and simulate the metal cutting processes, the issue of large deformation with elastoplasticity must be solved first. With the advent of advanced constitutive models along with improved numerical schemes, it will be possible to find the solution of many complex engineering problems involving large deformation and plasticity.

For the formulation of elastic-plastic theory at large deformation, Green-Naghdi's theory begins with the decomposition of the Green-Lagrange strain into the elastic part and plastic part as $E_{KL} = E^e_{KL} + E^p_{KL}$. On the other hand, Lee's theory [2] considers the multiplicative decomposition of deformation gradient F into an elastic F^e and plastic part F^p as $F = F^e F^p$. These two theories are formulated on the basis of fundamental laws of continuum mechanics and rigorously derived constitutive relations. According to [2], it is found that Green-Naghdi's theory is more flexible since it can be applied to either isotropic or anisotropic material and the computing procedure involved is relatively straightforward.

Since it is impossible to solve the real engineering problems involving large deformation plasticity in closed form except for the simplest ones, one has to use certain numerical methods to implement those theories of plasticity. Some researchers have used FEM to solve large strain deformation plasticity, such as Chiou [3], Lee [4], and Hu [5].

Over the past two decades the meshless methods have attracted much attention owing to their advantages in adaptivity, higher degree of continuity in the solution field, and capability to handle moving boundary and changing geometry. In the meshless method, the concept of an element is eliminated. The model geometry consists of a distribution of nodes over the domain, and the approximate solution is constructed entirely based on these nodes. Consequently, the nodal connectivity in the meshless method is much more flexible. Each node, through the use of

a localized weight function, is always connected to the nodes that are nearby, and there is no limitation on the number of other nodes a node can be connected to directly. This makes the model building in meshless methods much simpler than in FEM: adaptivity for meshless methods is easily achieved by adding or removing nodes without the troublesome remeshing of elements.

Several versions of meshless methods have been developed [1]. Most development in meshless methods to date has been focused mostly on linear elastic fracture mechanics. Research in large deformation elastoplastic fracture mechanics using meshless methods is only currently gaining attention. In [6], Belytschko proposed a 3D element-free Galerkin method intended for dynamic problems with geometric and material nonlinearities solved with explicit time integration. In [7], Rossi applied a modified element-free Galerkin method large deformation processes. The proposed EFG method enables the direct imposition of the essential boundary conditions. The plasticity model assumed a multiplicative decomposition of the deformation gradient into an elastic and plastic part and considered an elastoplastic constitutive relation that accounted for nonlinear isotropic hardening. In [8] and [9], Chen and his co-workers formulated the governing equations for rate-independent large strain plasticity in the framework of meshless method and used the method to simluate high-speed impact.

The methods cited above are "meshless" only in terms of the interpolation or approximation of the field or boundary variables, as compared to the usual boundary element method (BEM) or finite element method (FEM), but still have to use a background mesh to integrate a weak form over the problem domain or the boundary. The requirement of a background mesh for integration makes these methods not truly meshless.

The authors have developed a meshless integral method for linear elasticity [1] and extended it to elastoplasticity for small deformation [10]. This method is a truly meshless method and does not require a background mesh for integration. In the present work, we extend the meshless integral method to large deformation elastoplasticity. The governing integral equation is obtained from the weak form of large deformation elastoplasticity over a local sub-domain based on Green-Naghdi's theory, and the moving least-squares approximation is used for meshless function approximation. The constitutive law is a J_2 elastoplastic constitutive law that relates the Green-Lagrange strain to the second Piola-Kirchhoff stress based on von Mises yielding criterion with isotropic hardening. The fixed point iteration is used to solve the

nonlinear equations because of large deformation elastoplasticity. Because few closed-form solutions for realistic large deformation elastoplastic engineering problems are available, we compare the results from the meshless method with FEM results. The large deformation elastoplastic meshless integral method based on the regularized integral equation is accurate and robust, and appears remarkably promising. For all numerical tests, the meshless results are in excellent or satisfactory agreement with FEM solutions or hand calculation results.

The structure of this paper is as follows: in Section 4.3, the regularized local boundary integral equation is derived. The subtraction method is used to remove the strong singularity that is present in the local boundary integral equation. In Section 4.4, large deformation elastoplastic constitutive equations are presented. Section 4.5 describes the moving least-squares approximation (MLSA). The meshless implementation of the regularized boundary integral equation using MLSA is presented in Section 4.6. Section 4.7 discusses the treatment of weak singularity along with imposition of essential and natural boundary condition. The solution algorithm is given in Section 4.8. Numerical examples are presented in Section 4.9 to assess the accuracy and effectiveness of the method. Discussion and conclusions from this study are given in Section 4.10.

4.3 Regularized Local Boundary Integral Equation Using Subtraction Method

Consider a body in an initial state for a given load increment as shown in Figure 4.1; the domain of the body in the initial state is denoted by Ω_0 and called initial configuration (undeformed configuration). Undeformed configuration is only relative and designates the configuration with respect to which we measure deformation. In describing the motion of the body and deformation, we use the initial configuration as the *reference configuration*. The significance of the reference configuration lies in the fact that motion is defined with respect to this configuration. On the other hand, the domain of the *current configuration* of the body is denoted by Ω, and this is also called deformed configuration. The vector **X** for a given material point in the reference configuration does not change with time, and **X** is called Lagrangian coordinates; **x,** which describes the material point in the current configuration, changes with time, and it is called Eulerian coordinates. In this research, the rectangular Eulerian coordinates, x_k (k=1, 2) and Lagrangian coordinates, X_K are employed. We use subscripts I and J to indicate

that the variable of interest is referred to the reference state, and use subscripts i and j to indicate that the variable of interest is referred to the current state.

In this research, an updated Lagrangian formulation is derived. It means: (1) during each load increment, the state variables are defined with respect to the state at the start of this load increment (reference configuration); (2) at the end of this load increment, the state variables are updated with respect to the state at the end of this load increment (current configuration), and the current configuration will be the reference configuration for next load increment.

Consider a large deformation elastoplastic body represented by a reference domain Ω_0 with boundary Γ_0. The governing elastoplasticity equations are as follows:

$$\begin{cases} \tau_{IJ,J}(\mathbf{X}) + \rho_0 f_I(\mathbf{X}) = 0 \\ E_{IJ}(\mathbf{X}) = \dfrac{1}{2}(u_{I,J}(\mathbf{X}) + u_{J,I}(\mathbf{X}) + u_{K,I}(\mathbf{X})u_{K,J}(\mathbf{X})), \qquad \forall \mathbf{X} \in \Omega_0 \\ dT_{IJ}(\mathbf{X}) = C^e_{IJKL} dE_{kl}(\mathbf{X}) + dT^{cor}_{IJ}(\mathbf{X}) \end{cases} \tag{1}$$

where $\boldsymbol{\tau}$ (bold face denotes vectors or tensors) is the stress ($\tau_{IJ} = G^I_r \tau^{rJ} = G^I_r (T_{LJ} \dfrac{\partial x_r}{\partial X_L})$, T_{LJ} is the second Piola-Kirchhoff stress, **G** is the transformation tensor between the reference configuration and the current configuration, $\dfrac{\partial x_r}{\partial X_L}$ is the deformation gradient, **E** is the Green strain tensor associated with the displacement $\mathbf{u}$, ρ_0 is the mass density in the reference configuration Ω_0, f_I is the body force per unit mass, C^e_{IJKL} is the liner elastic stiffness matrix, and dT^{cor}_{IJ} is the correction term because of nonlinearities in either geometry or material. An index I following a comma designates partial differentiation with respect to X_I, and repeated indices indicate summation over the dimensionality of the problem. The essential and the natural boundary conditions on the boundary Γ are respectively:

$$u_I = \bar{u}_I \text{ on } \Gamma_u \tag{2}$$

$$t_I = \sigma_{IJ} n_J = \bar{t}_I \text{ on } \Gamma_t \tag{3}$$

here, $\bar{\mathbf{u}}$ represents the prescribed displacement on Γ_u, $\bar{\mathbf{t}}$ represents the prescribed traction on Γ_t; $\mathbf{n}$ is the outward unit normal to the boundary; $\Gamma_t \cup \Gamma_u = \Gamma$ and $\Gamma_t \cap \Gamma_u = \varnothing$.

The weak form of (1) over a local domain Ω_0 defined over the reference state is:

$$\int_{\Omega_0^{\langle a\rangle}} \left(\tau_{IJ,J}(\mathbf{X}) + \rho_0 f_I(\mathbf{X})\right) g_I\left(\mathbf{X}, \mathbf{Y}^{\langle a\rangle}\right) d\Omega_0(\mathbf{X}) = 0 \tag{4}$$

where the notation $<\cdot>$ in this paper is used to denote a node (e.g., <a> indicates node a, <b> indicates node b, etc.) in order to reserve the usual subscripts, I, J, etc., for denoting degree of freedom (DOF) components, $\Omega_0^{<a>} \subset \Omega_0$ is a sub-domain related to node <a>, $\mathbf{Y}^{<a>}$ is the position vector of node a which is also called a source point, $\mathbf{X}$ is the integration or field point which may or may not coincide with a node, and g_I is the test function. In the following, the functional dependence on $\mathbf{X}$, i.e. "$(\mathbf{X})$", will be dropped for brevity when no ambiguity is caused. In this work, following [1], we use a special test function defined by

$$g_I\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) = \tilde{u}^*_{JI}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) e_J\left(\mathbf{Y}^{<a>}\right) \tag{5}$$

where e_I represents the j-th component of a unit force vector, $\tilde{u}^*_{JI}$ is the special test function, given by [1]:

$$\tilde{u}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) = \frac{1}{8\pi\mu(1-\bar{\nu})}\left\{\left[(4\bar{\nu}-3)\ln\frac{r^{<a>}}{h_s^{<a>}} - \frac{5-4\bar{\nu}}{2(3-4\bar{\nu})}\left(1-\frac{r^{<a>2}}{h_s^{<a>2}}\right)\right]\delta_{IJ} + \left(1-\frac{r^{<a>2}}{h_s^{<a>2}}\right)\frac{r_I^{<a>} r_J^{<a>}}{r^{<a>2}}\right\} \tag{6}$$

The associated traction is:

$$\begin{aligned}\tilde{t}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) = -\frac{1}{4\pi(1-\bar{\nu})}\Bigg\{ &\left[\frac{1}{r^{<a>}}\frac{\partial r^{<a>}}{\partial n^{<a>}}(1-2\bar{\nu}) - \frac{r_K^{<a>} n_K^{<a>}}{(3-4\bar{\nu})h_s^{<a>2}}\right]\delta_{IJ} + 2\frac{\partial r^{<a>}}{\partial n^{<a>}}\frac{r_I^{<a>} r_J^{<a>}}{r^{<a>3}} \\ &- \frac{1-2\bar{\nu}}{r^{<a>2}}\left(r_I^{<a>} n_J^{<a>} - r_J^{<a>} n_I^{<a>}\right) + \frac{3r_I^{<a>} n_J^{<a>} - r_J^{<a>} n_I^{<a>}}{(3-4\bar{\nu})h_s^{<a>2}}\Bigg\}\end{aligned} \tag{7}$$

where $r^{<a>}$ is the distance from $\mathbf{Y}^{<a>}$ to $\mathbf{X}$; $r_I^{<a>} = X_I - Y_I^{<a>}$; $\mathbf{n}^{<a>}$ is the outward unit normal to the boundary $\partial\Omega_0^{<a>}$ at $\mathbf{X}$; $\bar{\nu} = \nu$ and $\bar{E} = E$ for plane strain, or $\bar{\nu} = \nu/(1+\nu)$ and $\bar{E} = E(1-\bar{\nu}^2)$ for plane stress; E is the Young's modulus, ν is the Poisson's ratio, and $\mu = \bar{E}/(2(1+\bar{\nu}))$ is the shear modulus. $h_s^{<a>}$ is the radius of the local sub-domain.

The special test function $\tilde{u}^*_{IJ}$ has the property that it vanishes on the boundary of a spherical $\Omega_0^{<a>}$. With $\tilde{u}^*_{IJ}$ as the test function, application of integration by parts to (4) twice leads to [1]:

$$
\begin{aligned}
&- \int_{\Omega_0^{<a>}} \delta\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) u_I(\mathbf{X}) d\,\Omega(\mathbf{X}) + \int_{\partial\Omega_0^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) T^0_J(\mathbf{X}) d\,\Gamma_0(\mathbf{X}) \\
&- \int_{\partial\Omega_0^{<a>}} \tilde{t}^*_{IJ}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) u_J(\mathbf{X}) d\,\Gamma_0(\mathbf{X}) - \int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ,K}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) T^{cor}_{LK}(\mathbf{X}) \delta_{rJ} \frac{\partial x_r}{\partial X_L} d\Omega(\mathbf{X}) \\
&- \int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ,K}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) \delta_{RJ} \Delta u_{R,L} C^e_{LKMN} u_{M,N} d\Omega(\mathbf{X}) + \int_{\partial\Omega_0^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) T^0_{LK}(\mathbf{X}) \Delta u_{J,L}(\mathbf{X}) n_K d\,\Gamma_0(\mathbf{X}) \quad (8) \\
&+ \int_{\partial\Omega_0^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) \Delta T_{LK}(\mathbf{X}) (\delta_{JL} + \Delta u_{J,L}(\mathbf{X})) n_K d\,\Gamma_0(\mathbf{X}) \\
&+ \int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) \rho_0 f_J(\mathbf{X}) d\,\Omega(\mathbf{X}) = 0
\end{aligned}
$$

where $\partial\Omega_0^{<a>}$ is the boundary of $\Omega_0^{<a>}$, T^0_J is the J-th component of the traction at reference state, Δu is the displacement increment from the reference state to the current state, T^0_{LK} is the second Piola-Kirchhoff stress at the reference state, and ΔT_{LK} is the second Piola-Kirchhoff stress increment from the reference state to the current state. Equation (8) in its current form cannot be used directly in numerical calculation because when $\mathbf{Y}^{<a>}$ is a boundary node, the equation contains strong singularity (1/r type in line integral) in the traction term $\tilde{t}^*_{IJ}$. For the local integral equation approach to be a valid numerical method, the strong singularity must be handled appropriately.

The subtraction technique is employed in the present study to remove the strong singularity, for which a limiting process is performed as discussed below. For simplicity in implementation, the local sub-domain is always chosen, in the reference state, as a sphere or part of a sphere centered on a node. If node $\mathbf{Y}^{<a>}$ is an interior node, $h_s^{<a>}$ is selected such that $\Omega_0^{<a>}$ stays fully inside Ω_0. If $\mathbf{Y}^{<a>}$ is a boundary node, then $\Omega_0^{<a>}$ is the intersection of Ω_0 and a sphere $\Omega_0^{'<a>}$ of radius $h_s^{<a>}$, centered at the boundary node, and the boundary $\partial\Omega_0^{<a>}$ is the union of the part of $\partial\Omega'^{\langle a\rangle}$ inside Ω_0 and the part of $\partial\Omega_0^{<a>}$ inside $\Omega^{'<a>}$, as illustrated in Figure 4.2. A further modification is the exclusion of a tiny sphere Ω_Δ of radius Δ (which later tends to zero) centered on $\mathbf{Y}^{<a>}$. Figure 4.3 shows schematically this modification.

We now decompose the boundary $\partial\Omega_0^{<a>}$ into the following sections:

$$
\partial\Omega_0^{<a>} = C^{<a>} \cup C^{<a>}_\Delta \cup \Gamma^{<a>} \quad (9)
$$

$$\Gamma^{<a>} = \Gamma_u^{<a>} \cup \Gamma_t^{<a>} \tag{10}$$

where $C^{<a>}$ is the circular part of $\partial\Omega_0{}^{<a>}$ of radius $h_s^{<a>}$, C_Δ is the circular part of $\partial\Omega_\Delta$ of radius Δ, $\Gamma_u^{<a>}$ is the section of $\Gamma^{<a>}$ where the displacement is prescribed and $\Gamma_t^{<a>}$ the section where the traction is prescribed. For interior nodes, $\Gamma^{<a>}$ is zero, and $C^{<a>}$ is a full circle.

Using subtraction method, we obtain, in the limit of $\Delta \to 0$, the following:

$$\begin{aligned}
&-\int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ,K}\left(\mathbf{X},\mathbf{Y}^{<a>}\right) T^{cor}_{LK}(\mathbf{X})\delta_{rJ}\frac{\partial x_r}{\partial X_L}d\Omega(\mathbf{X}) \\
&-\int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ,K}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)\delta_{RJ}\Delta u_{R,L} C^e_{LKMN} u_{M,N} d\Omega(\mathbf{X}) \\
&+\int_{\partial\Omega_0^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right) T^0_{LK}(\mathbf{X})\Delta u_{J,L}(\mathbf{X}) n_K d\Gamma(\mathbf{X}) \\
&+\int_{\partial\Omega_0^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)\Delta T_{LK}(\mathbf{X})(\delta_{JL}+\Delta u_{J,L}(\mathbf{X}))n_K d\Gamma(\mathbf{X}) \\
&+\int_{\Gamma^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right) T^0_J(\mathbf{X})d\Gamma(\mathbf{X}) - \int_{C^{<a>}} \tilde{t}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)(u_J(\mathbf{X})-u_J^{<a>})d\Gamma(\mathbf{X}) \\
&-\int_{\Gamma^{<a>}} \tilde{t}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)(u_J(\mathbf{X})-u_J^{<a>})d\Gamma(\mathbf{X}) + \int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)\rho_0 f_J(\mathbf{X})d\Omega(\mathbf{X}) = 0
\end{aligned} \tag{11}$$

The integration of $\tilde{t}^*_{IJ}(\mathbf{X},\mathbf{Y}^{<a>})$ over $C^{<a>}$ can be obtained in closed form:

$$\int_{C^{<a>}} \tilde{t}^*_{IJ}(\mathbf{X},\mathbf{Y}^{<a>})d\Gamma = -\alpha_{IJ}^{<a>} \tag{12}$$

with

$$-[\alpha_{IJ}^{<a>}] = \begin{bmatrix} -\frac{\theta}{2\pi} - \frac{\sin 2\theta_2 - \sin 2\theta_1}{2\pi(3-4\bar{\nu})} & \frac{\cos 2\theta_2 - \cos 2\theta_1}{2\pi(3-4\bar{\nu})} \\ \frac{\cos 2\theta_2 - \cos 2\theta_1}{2\pi(3-4\bar{\nu})} & -\frac{\theta}{2\pi} + \frac{\sin 2\theta_2 - \sin 2\theta_1}{2\pi(3-4\bar{\nu})} \end{bmatrix} \tag{13}$$

Here $\theta_2 - \theta_1 = \theta$ is the internal boundary angle subtended by material at $\mathbf{Y}^{\langle a\rangle}$ on the boundary, as shown in Figure 4.4. Two special cases are worth noting. For an interior node, $\theta = 2\pi$; while for a boundary node where the boundary is smooth, $\theta = \pi$.

Substituting the above expressions into (11) leads to:

$$\alpha_{IJ}^{<a>}u_J^{<a>} = -\int_{\Omega_0^{<a>}} \tilde{u}_{IJ,K}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)T_{LK}^{cor}(\mathbf{X})\delta_{RJ}\frac{\partial x_r}{\partial X_L}d\Omega(\mathbf{X})$$
$$-\int_{\Omega_0^{<a>}} \tilde{u}_{IJ,K}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)\delta_{RJ}\Delta u_{R,L}C_{LKMN}^e u_{M,N}d\Omega(\mathbf{X})$$
$$+\int_{\partial\Omega_0^{<a>}} \tilde{u}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)T_{LK}^0(\mathbf{X})\Delta u_{J,L}(\mathbf{X})n_K d\Gamma(\mathbf{X}) + \int_{\Omega_0^{<a>}} \tilde{u}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)\rho_0 f_J(\mathbf{X})d\Omega \qquad (14)$$
$$+\int_{\partial\Omega_0^{<a>}} \tilde{u}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)\Delta T_{LK}(\mathbf{X})(\delta_{JL}+\Delta u_{J,L}(\mathbf{X}))n_K d\Gamma(\mathbf{X}) + \int_{\Gamma_u^{<a>}+\Gamma_t^{<a>}} \tilde{u}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)T_J^0(\mathbf{X})d\Gamma$$
$$-\int_{C^{<a>}} \tilde{t}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_J(\mathbf{X})d\Gamma - \int_{\Gamma_u^{<a>}+\Gamma_t^{<a>}} \tilde{t}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)(u_J(\mathbf{X})-u_J^{<a>})d\Gamma$$

In (14), the subtraction method has been used in the last term of the right hand side. When the field point $\mathbf{X}$ approaches the source node $\mathbf{Y}^{\langle a\rangle}$, $u_I(x)-u_I^{<a>}$ tends to zero which removes the strong singularity and makes the integral numerically integrable. All other terms in (14) are regular or weakly singular for which special integration quadrature gives convergent and accurate results. The regularized Equation (14) holds for any source node $\mathbf{Y}^{\langle a\rangle}$, either inside the domain or on the boundary. In the current meshless integral method, both boundary and interior source nodes are used, and the moving least-squares approximation is employed for approximating the solution field, as presented later.

4.4 Constitutive Equation for Elastoplasticity with Large Deformation

In this research, we use Green-Naghdi's theory, associated flow theory, von Mises yield criterion, and isotropic strain hardening to model the material behavior. Green-Naghdi's theory [2] begins with the decomposition of the Green-Lagrange strain increment into the elastic and plastic parts as $dE_{KL} = dE_{KL}^e + dE_{KL}^p$.

The von Mises yield criterion has the following form:

$$f = \frac{1}{2}S_{IJ}S_{IJ} - \kappa = \frac{1}{2}S_{IJ}S_{IJ} - \frac{1}{3}\sigma_e{}^2 = 0 \qquad (15)$$

f is the yield function constructed in the space of second Piola-Kirchhoff stress, S_{IJ} is the component of deviatoric stress of second Piola-Kirchhoff stress, and σ_e is the effective stress, κ is the material parameter characterizing hardening effect. Effective stress

$\sigma_e = Y_0 + \varepsilon_p H_p$, Y_0 is the initial yield stress, H_p is the hardening modulus, and ε_p is the effective plastic strain. H_p is given by the following equation

$$H_p = \frac{E_T}{1 - E_T/E_Y}, \tag{16}$$

where E_T is elastoplastic tangent modulus and E_Y is the elastic Young's modulus from uniaxial tension test.

The plastic strain increment dE_{IJ}^p is expressed by an associated flow rule as:

$$dE_{IJ}^p = \frac{L}{H}\frac{\partial f}{\partial \sigma_{IJ}} = \frac{L}{H} S_{IJ} \tag{17}$$

where L is the loading criteria and given by the following equation:

$$\begin{aligned} L &= \left\{\frac{\partial f}{\partial \{T\}}\right\}^T [C]\{dE\} \\ &= \frac{E}{1+v}[S_{11}dE_{11} + S_{22}dE_{22} + S_{33}dE_{33} + 2S_{12}dE_{12}] \end{aligned} \tag{18}$$

[C] is elastic stiffness matrix which works for both plane strain and plane stress.

The positive scalar H is expressed as:

$$H = \frac{2}{3}\sigma_e^{\,2}\left(\frac{2}{3}\frac{d\sigma_e}{d\varepsilon_p} + \frac{E}{1+v}\right) \tag{19}$$

Using Equation (18) and (19), we obtain the plastic strain increment:

$$\left\{dE_{IJ}^{\,p}\right\} = \frac{L}{H}\begin{Bmatrix} S_{11} \\ S_{22} \\ S_{33} \\ 2S_{12} \end{Bmatrix} \tag{20}$$

With dE_{IJ}^p, the increment of effective plastic strain is computed as follows:

$$d\varepsilon_p = C\sqrt{dE_{IJ}^{\,p} dE_{IJ}^{\,p}} = \sqrt{\frac{2}{3}}\frac{L}{H}\sqrt{S_{IJ}S_{IJ}} = \sqrt{\frac{2}{3}}\frac{L}{H}\sqrt{\frac{2}{3}}\sigma_{eq} = \frac{2}{3}\sigma_{eq}\frac{L}{H} \tag{21}$$

where $\sigma_{eq} = \sqrt{\frac{3}{2}S_{IJ}S_{IJ}}$ is the equivalent stress of current stress state.

The constitutive equations are summarized as follows:

a. Elastic region ($f(T_{IJ}) < 0$)

$$dT_{IJ} = 2\mu dE_{IJ} + \lambda \frac{dT_{KK}}{2\mu + 3\lambda}\delta_{IJ} + [-\frac{1}{2}(\delta_{IK}\sigma_{JL} + \delta_{JK}\sigma_{IL} + \delta_{IL}\sigma_{JK} + \delta_{JL}\sigma_{IK}) + \sigma_{IJ}\delta_{KL}]dE_{KL} \quad (22)$$

Where λ and μ are Lame constants.

b. Plastic region (loading, $f(T_{IJ}) = 0, L \geq 0$)

$$\begin{aligned} dT_{IJ} &= 2\mu dE_{IJ} + \lambda dE_{KK}\delta_{IJ} - \frac{9\mu^2}{3\mu + H_p}\frac{S_{KL}dE_{KL}S_{IJ}}{\sigma_e^2} \\ &+ [-\frac{1}{2}(\delta_{IK}\sigma_{JL} + \delta_{JK}\sigma_{IL} + \delta_{IL}\sigma_{JK} + \delta_{JL}\sigma_{IK}) + \sigma_{IJ}\delta_{KL}]dE_{KL} \end{aligned} \quad (23)$$

c. Plastic region (unloading, $f(T_{IJ}) = 0, L < 0$)

$$dT_{IJ} = 2\mu\, dE_{IJ} + \lambda dE_{KK}\delta_{IJ} + [-\frac{1}{2}(\delta_{IK}\sigma_{JL} + \delta_{JK}\sigma_{IL} + \delta_{IL}\sigma_{JK} + \delta_{JL}\sigma_{IK}) + \sigma_{IJ}\delta_{KL}]dE_{KL} \quad (24)$$

4.5 The Moving Least-squares Approximation

In the finite element method, the coupling between the nodes is accomplished through the use of shape functions, defined locally over each element, which interpolate the solution field from nodal values. For a meshless method, the absence of elements excludes the use of such shape functions and therefore, a different local approximation scheme based on nodal values but independent of any elements needs to be devised. In this work, we have chosen to exploit the non-interpolative moving least-squares approximation (MLSA) scheme because of its high accuracy and the ease with which it can be extended to n-dimensional problems.

Consider a domain Ω that contains n nodes:

$$\mathbf{Y}^{<a>} = \begin{cases} [Y_1^{<a>}, Y_2^{<a>}]^T & \text{in 2 - D} \\ [Y_1^{<a>}, Y_2^{<a>}, Y_3^{<a>}]^T & \text{in 3 - D} \end{cases}, \; a = 1 \cdots n \quad (25)$$

For any of these nodes $\mathbf{Y}^{\langle a \rangle}$, following [1][10][13], we define a ***support domain*** for node $\mathbf{Y}^{\langle a \rangle}$, which is a sphere (3D) or disk (2D) centered on $\mathbf{Y}^{\langle a \rangle}$ with a radius $\ell_w^{\langle a \rangle}$. A weight function $w^{<a>}$ is a continuous function that is positive in the support domain and zero outside, i.e.

$$\begin{cases} w^{\langle a \rangle}(\mathbf{X}) \geq 0 & \text{if } \left|\mathbf{X} - \mathbf{Y}^{\langle a \rangle}\right| \leq \ell_w^{\langle a \rangle} \\ w^{\langle a \rangle}(\mathbf{X}) = 0 & \text{if } \left|\mathbf{X} - \mathbf{Y}^{\langle a \rangle}\right| > \ell_w^{\langle a \rangle} \end{cases} \quad (26)$$

As introduced previously, the ***sub-domain*** $\Omega^{\langle a \rangle}$ for node $\mathbf{Y}^{\langle a \rangle}$, located entirely inside Ω, is a sphere or part of a sphere centered on $\mathbf{Y}^{\langle a \rangle}$ with a radius $h_s^{<a>}$. Figure 4.5 illustrates the meaning of local sub-domain and support domain.

Two other frequently used concepts are the domain of definition and the domain of influence. The ***domain of definition*** of a point $\mathbf{x}$ is the set of all nodes whose weight functions are non-zero at **X**, while the ***domain of influence*** of a node $\mathbf{Y}^{\langle a \rangle}$ is the set of all nodes whose weight functions are non-zero in some part or all of the sub-domain of node $\mathbf{Y}^{\langle a \rangle}$. The domain of definition and the domain of influence are convenient terms in the description of MLSA and local boundary integrals, and are illustrated schematically in Figure 4.6.

The moving least-squares ***approximant*** u_h to a function **u** is defined by:

$$\mathbf{u}_h(\mathbf{x}) = \mathbf{p}^T(\mathbf{X})\mathrm{c}(\mathbf{X}), \ \forall \mathbf{X} \in \Omega \tag{27}$$

The two vectors $\mathbf{p}$ and $\mathbf{c}$ are both functions of the spatial coordinates: $\mathbf{X} = [X_1, X_2]^T$ in 2D or $\mathbf{X} = [X_1, X_2, X_3]^T$ in 3D. $\mathbf{p}$ is a complete monomial basis of *m* terms (e.g., in 2D, m=3 for a linear basis, and 6 for a quadratic basis), $\mathbf{c}$ is a coefficient vector which is determined by minimizing a weighted discrete L_2-norm:

$$\mathbf{J}(\mathbf{X}) = \sum_{a=1}^{N_x} (w^{<a>}(\mathbf{X})[\mathbf{p}^T(\mathbf{Y}^{<a>})\mathbf{c}(\mathbf{X}) - \hat{\mathbf{u}}^{<a>}]^2) \tag{28}$$

where $\hat{\mathbf{u}}^{\langle a \rangle}$ is the fictitious nodal displacement that approximates the value of $\mathbf{u}$ at node $\mathbf{Y}^{\langle a \rangle}$, and the upper limit of summation, N_x, is the total number of nodes in the domain of definition of point **X**. The matrices $\mathbf{P}$, $\mathbf{W}$ and $\hat{\mathbf{u}}$ are defined by

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}^T(\mathbf{Y}^{<1>}) \\ \mathbf{p}^T(\mathbf{Y}^{<2>}) \\ \cdots \\ \mathbf{p}^T(\mathbf{Y}^{<N_x>}) \end{bmatrix}_{N_x \times m} \tag{29}$$

$$\mathbf{W}(\mathbf{x}) = \begin{bmatrix} w^{<1>}(\mathbf{X}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w^{<Nx>}(\mathbf{X}) \end{bmatrix}_{N_x \times N_x} \tag{30}$$

$$\hat{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}^{<1>} \\ \hat{\mathbf{u}}^{<2>} \\ \vdots \\ \hat{\mathbf{u}}^{<N_x>} \end{bmatrix} \tag{31}$$

Minimization of (28) leads to:

$$\mathbf{u}_h(\mathbf{X}) = \mathbf{\Phi}^T(\mathbf{X})\hat{\mathbf{u}} = \sum_{a=1}^{N_x} \varphi^{<a>}(\mathbf{X})\hat{\mathbf{u}}^{<a>} \tag{32}$$

where:

$$\mathbf{\Phi}^T(\mathbf{X}) = \mathbf{p}^T(\mathbf{X})\mathbf{A}^{-1}(\mathbf{X})\mathbf{B}(\mathbf{X}) \tag{33}$$

$$\text{or } \varphi^{<a>}(\mathbf{X}) = \sum_{b=1}^{N_x} p_b(\mathbf{X})\left[\mathbf{A}^{-1}(\mathbf{X})\mathbf{B}(\mathbf{X})\right]_{ba} \tag{34}$$

$$\mathbf{A}(\mathbf{X}) = \mathbf{P}^T\mathbf{W}(\mathbf{X})\mathbf{P} = \mathbf{B}(\mathbf{X})\mathbf{P} = \sum_{a=1}^{N_x} w^{<a>}(\mathbf{X})\mathbf{p}(\mathbf{Y}^{<a>})\mathbf{p}^T(\mathbf{Y}^{<a>}) \tag{35}$$

$$\mathbf{B}(\mathbf{X}) = \mathbf{P}^T\mathbf{W}(\mathbf{X}) = \left[w^{<1>}(\mathbf{X})\mathbf{p}(\mathbf{Y}^{<1>}) \;\; w^{<2>}(\mathbf{X})\mathbf{p}(\mathbf{Y}^{<2>}) \;\cdots\; w^{<N_x>}(\mathbf{X})\mathbf{p}(\mathbf{Y}^{<N_x>})\right] \tag{36}$$

The MLSA for a function exists only when $\mathbf{A}(\mathbf{X})$ is non-singular. A necessary condition for a well-defined MLSA is that for each sample point $\mathbf{X} \in \Omega_0$ (a node or a quadrature point), at least *m* weight functions are non-zero and the nodes in the domain of definition of $\mathbf{X}$ are not arranged in a degenerate pattern (such as on a straight line). In MLSA, the shape function related to node $\mathbf{Y}^{\langle a \rangle}$ is $\varphi^{<a>}(\mathbf{X})$. The size of the support domain should be large enough to ensure the coupling between a minimum set of nodes, but small enough to capture local variations.

The influence of the choices of the basis functions and the weight functions on the behavior and the quality of the shape function has been discussed in [1]. In this work, we use linear, quadratic monomial basis, and spline weight functions, defined as follows:

Spline: $$w^{\langle a\rangle}(\mathbf{X})=\begin{cases}1-6\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^2+8\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^3-3\left(\dfrac{r^{\langle a\rangle}}{\ell_w^{\langle a\rangle}}\right)^4, & 0\le r^{\langle a\rangle}\le \ell_w^{\langle a\rangle}\\ 0, & r^{\langle a\rangle}>\ell_w^{\langle a\rangle}\end{cases} \tag{37}$$

here $r^{<a>}$ is the distance from node $\mathbf{Y}^{\langle a\rangle}$ to point $\mathbf{X}$, $\ell_w^{\langle a\rangle}$ is the size of support domain. The weight function has only one parameter, the size of support $\ell_w^{\langle a\rangle}$, which makes its use simpler. It is noted that MLSA is non-interpolative, and there is a difference between the nodal value of the MLSA approximant $\mathbf{u_h}$ and the fictitious nodal displacement $\hat{\mathbf{u}}^{<a>}$. For brevity, the subscript h in $\mathbf{u_h}$ will be omitted in the remaining part of this paper.

4.6 Meshless Implementation

We now apply the MLSA to the integral Equation (14) to establish the meshless implementation. The shape function, as we have defined it, gives:

$$u_J(\mathbf{X})=\sum_{b=1}^{N_x}\varphi^{<b>}(\mathbf{X})\hat{u}_J^{<b>} \tag{38}$$

$$u_{J,K}(\mathbf{X})=\sum_{b=1}^{N_x}\varphi_{,K}^{<b>}(\mathbf{X})\hat{u}_J^{<b>} \tag{39}$$

$$\Delta u_{J,K}(\mathbf{X})=\sum_{b=1}^{N_x}\varphi_{,K}^{<b>}(\mathbf{X})\Delta\hat{u}_J^{<b>} \tag{40}$$

where N_x is the total number of nodes in the domain of definition of point $\mathbf{X}$.

The related traction term T_J is

$$T_J=\sigma_{IJ}(\mathbf{X})n_I(\mathbf{X}) \tag{41}$$

where (n_1, n_2) is the normal to the plane passing $\mathbf{X}$ over which the traction acts. For a node $\mathbf{Y}^{<b>}$, we define $\mathbf{N}$ and $\mathbf{B}^{\langle b\rangle}$ matrices as:

$$\mathbf{N}=\begin{bmatrix}n_1 & 0 & n_2\\ 0 & n_2 & n_1\end{bmatrix};\ \mathbf{B}^{<b>}=\begin{bmatrix}\varphi_{,1}^{<b>} & 0\\ 0 & \varphi_{,2}^{<b>}\\ \varphi_{,2}^{<b>} & \varphi_{,1}^{<b>}\end{bmatrix} \tag{42}$$

Combining (42) with (39) and (41), we can express the traction in terms of the shape functions as follows:

$$\begin{Bmatrix} t_1(\mathbf{X}) \\ t_2(\mathbf{X}) \end{Bmatrix} = \sum_{b=1}^{N_x} [NC^{ep}_{IJKL} B^{<b>}(\mathbf{Y}^{<a>})] \begin{Bmatrix} \hat{u}_1^{<b>} \\ \hat{u}_2^{<b>} \end{Bmatrix} \tag{43}$$

here C^{ep}_{IJKL} is the elastoplastic stiffness matrix.

With the above discretization and the boundary conditions that $u_J = \bar{u}_J$ on $\Gamma_u^{<a>}$ and $T_J = \overline{T}_J$ on $\Gamma_t^{<a>}$, Equation (14) becomes (there is a summation on b and j but not on a and i):

$$\sum_{b=1}^{N_y} (H_{IJ}^{<a,b>} \hat{u}_J^{<b>}) - \sum_{b=1}^{N_y} (L_{IJ}^{<a,b>} \hat{u}_J^{<b>}) + \alpha_{IJ}^{<a>} u_J^{<a>} = G_I^{<a>} \tag{44}$$

where $H_{IJ}^{<a,b>}$, $L_{IJ}^{<a,b>}$ and $G_I^{<a>}$ are:

$$\begin{aligned} H_{IJ}^{<a,b>} = &- \int_{\Gamma_u^{<a>}} \tilde{u}^*_{IK}(\mathbf{X}, \mathbf{Y}^{<a>}) [\mathbf{N}\mathbf{C}^{ep}\mathbf{B}^{<b>}(\mathbf{X})]_{KJ} d\Gamma + \int_{C^{<a>}} \tilde{t}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) \varphi^{<b>}(\mathbf{X}) d\Gamma, \\ &+ \int_{\Gamma_t^{<a>}} \tilde{t}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) (\varphi^{<b>}(\mathbf{X})) d\Gamma. \end{aligned} \tag{45}$$

$$L_{IJ}^{<a,b>} = \int_{\Gamma_t^{<a>}} \tilde{t}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) (\varphi^{<b>}(\mathbf{Y}^{<a>})) d\Gamma \tag{46}$$

$$\begin{aligned} G_I^{<a>} = &\int_{\Omega_0^{<a>}} b_J(\mathbf{X}) \tilde{u}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) d\Omega + \int_{\Gamma_t^{<a>}} \tilde{u}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) \overline{T}^0_J(\mathbf{X}) d\Gamma \\ &- \int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ,K}(\mathbf{X}, \mathbf{Y}^{<a>}) T^{cor}_{LK}(\mathbf{X}) \delta_{RJ} (\delta_{rL} + \Delta u_{r,L}) d\Omega(\mathbf{X}) \\ &- \int_{\Omega_0^{<a>}} \tilde{u}^*_{IJ,K}(\mathbf{X}, \mathbf{Y}^{<a>}) \delta_{RJ} \Delta u_{R,L} C^e_{LKMN} u_{M,N} d\Omega(\mathbf{X}) + \int_{\partial\Omega_0^{<a>}} \tilde{u}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) T^0_{LK}(\mathbf{X}) \Delta u_{J,L}(\mathbf{X}) n_K d\Gamma(\mathbf{X}) \\ &+ \int_{\partial\Omega_0^{<a>}} \tilde{u}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) \Delta T_{LK}(\mathbf{X}) (\delta_{JL} + \Delta u_{J,L}(\mathbf{X})) n_K d\Gamma(\mathbf{X}) - \int_{\Gamma_u^{<a>}} \tilde{t}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) (\bar{u}_J(\mathbf{X}) - \bar{u}_J(\mathbf{Y}^{<a>})) d\Gamma \end{aligned} \tag{47}$$

and the upper limit of summation, N_y, is the total number of nodes in the domain of influence of node $\mathbf{Y}^{<a>}$. The third to sixth terms of Equation (47) are the plastic terms because of the large deformation elastoplasticity.

In all numerical examples tested in this work, the body force term is zero.

4.7 Treatment for Weak Singularity and Imposition of Boundary Conditions

Owing to the subtraction technique, the singularity in the third integral of (45) as $\mathbf{X}$ approaches $\mathbf{Y}^{<a>}$ is cancelled by the term in (46), and similarly the seventh integral in (47) is also regularized. Even though the subtraction technique removes the strong singularity, the integrands in the first integral of (45) and the second integral in (47) still contain the weakly singular ln(r) term. The logarithmic singularity is integrable, but the accuracy of ordinary Legendre-Gauss integration is poor. We found that the special integration scheme for the logarithmic singularity [11], which is reproduced in Table 4.1 for completeness, achieves excellent numerical accuracy [1]. The remaining integrals of Equation (45) and Equation (47) are regular for which standard quadrature can be used with good accuracy.

In our numerical examples, the numbers of integration points were as follows: 8 integration points for any integral along a straight line, and 64 integration points for any integral over an sub-domain. For regularized integrals, the usual Legendre-Gauss integration was used. For integrals containing logarithmic singularity, the special integration of 8 integration points listed in Table 4.1 was used.

Appropriate boundary conditions need to be imposed in order to solve the simultaneous Equations (44). In meshless methods, imposing the essential (Dirichlet) boundary conditions is not as trivial as in the finite element method. Because MLSA is non-interpolative, the essential boundary condition does not take the form of prescribed value for the fictitious nodal displacement ($\hat{u}_I^{<a>} = \overline{u}_I^{<a>}$), but rather a constraint equation involving a linear combination of the fictitious nodal displacements in a neighborhood of the boundary node (i.e., $u_I^{\langle a\rangle} = \sum_{b=1}^{n} \varphi^{\langle b\rangle}\left(\mathbf{Y}^{\langle a\rangle}\right)\hat{u}_I^{\langle b\rangle} = \overline{u}_I^{\langle a\rangle}$). A number of techniques for the imposition of essential boundary conditions have been developed, including: (1) Collocation methods [12]; (2) Lagrange multiplier method [13]; (3) Penalty method [14][15]; (4) Nitsche's method [16]; (5) Coupled meshless-finite element method [17]; (6) Admissible approximation method [18]; (7) Method based on d'Alembert principle [19]; (8) Use of window or correction functions that vanish on the boundary [20]; (9) Discrete form of essential boundary conditions [21]; and (10) Displacement constraint equation method [22]. The advantages and disadvantages of each of these methods have been discussed briefly in [1].

Collocation methods for enforcing essential boundary conditions are defined as those methods in which conditions are enforced exactly at a discrete set of boundary nodes [23]. A

number of collocation methods have been developed. The direct collocation method [12] used the collocation condition

$$\hat{u}_I^{<a>} = \overline{u}_I^{<a>} \tag{48}$$

to replace the row of the discretized weak form equation corresponding to the degree of freedom with prescribed displacement $\overline{u}_I^{<a>}$. This is actually inconsistent with the assumption of MLSA since the fictitious nodal displacement $\hat{u}_I^{<a>}$ is generally not equal to the approximated displacement value.

A modified collocation method uses

$$u_I^{\langle a\rangle} = \sum_{b=1}^{n} \varphi^{\langle b\rangle}\left(\mathbf{Y}^{\langle a\rangle}\right)\hat{u}_I^{\langle b\rangle} = \overline{u}_I^{\langle a\rangle} \tag{49}$$

as the collocation condition which was shown to yield more accurate results [23]. Wagner and Liu [24] developed a corrected collocation method which restores the consistency of the weak form and enhances convergence. Wu and Plesha [25] proposed a boundary flux collocation method to enforce the boundary conditions exactly.

Generally, there are two types of discretization in meshless methods: (1) Local collocation over multiple local domains, which is employed in [26][27][28][29]; (2) Galerkin based method over the global domain, which is used in EFG [30][12][31][21], clouds [20], RKPM [32][33][34][35], etc. For local collocation based discretization, each equation is obtained by applying the weak form over a particular local domain, and the weak form needs to be applied n times for a problem with n DOFs. Consequently, the collocation method with (49) can be used easily and directly to impose essential boundary conditions, because each of the system equations is independent of the rest, and replacing the equation corresponding to a constrained DOF by (49) will not cause any inconsistency in the weak form. For the Galerkin based discretization, the n system equations are obtained by applying the weak form over the global domain once, and therefore all equations must hold simultaneously in order to maintain consistency in the weak form. Replacing a row in the matrix equation by (49), which contains a linear combination of DOFs rather than dictating the value of the single constrained DOF, sacrifices the consistency of the weak form and compromises the accuracy of the solution.

The current meshless integral method utilizes the local collocation based discretization (over multiple sub-domains), and the collocation method (49) can be directly used to impose

essential boundary conditions. Since the system equations are obtained by applying the integral Equation (14) to each source node over a local sub-domain, for a DOF with essential boundary condition, we simply use the essential boundary condition $u_I = \bar{u}_I$ rather than applying the integral Equation (14). This is equivalent to replacing the governing equation corresponding to the DOF with essential boundary condition (49). Our numerical tests presented in the numerical examples section show that the collocation method (49) for imposing essential boundary conditions works very well with the meshless integral method.

For the natural boundary condition $T_I = \overline{T}_I$, no special treatment is needed. The prescribed traction is incorporated in the system governing equation (47) and directly used in the second integral.

After the boundary conditions are imposed, the governing equations can be written as

$$\sum_{b=1}^{N_y} (K_{IJ}^{<a,b>} \hat{u}_J^{<b>}) = R_I^{<a>} \tag{50}$$

where

$$K_{IJ}^{<a,b>} = \begin{cases} H_{IJ}^{<a,b>} - L_{IJ}^{<a,b>} + \alpha_{IJ}^{<a>} \varphi^{<b>}(\mathbf{Y}^{<a>}) & \text{when } u_I^{<a>} \text{ is unconstrained} \\ \varphi^{<b>}(\mathbf{Y}^{<a>}) \delta_{IJ} & \text{when } u_I^{<a>} = \bar{u}_I^{<a>} \end{cases} \tag{51}$$

$$R_I^{<a>} = \begin{cases} G_I^{<a>} & \text{when } u_I^{<a>} \text{ is unconstrained} \\ \bar{u}_I^{<a>} & \text{when } u_I^{<a>} = \bar{u}_I^{<a>} \end{cases} \tag{52}$$

and the upper limit of summation, N_y, is the total number of nodes in the domain of influence of node $\mathbf{Y}^{<a>}$.

4.8 Solution Algorithm for Elastoplasticity with Large Deformation

The governing equations are nonlinear due to large deformation elastoplasticity. In this work, we will use the fixed point iteration to solve the governing equations which has the following advantages: the derivative of the stiffness matrix is not needed, and the implementation is relatively easy. With this algorithm, for each load increment, the stiffness matrix is computed only once in the first iteration.

Throughout this section, we use $\{\text{var}\}_{[m]}^{(i)}$ to denote a variable $\{\text{var}\}$(displacement, stress or strain) for m-th load increment at i-th iteration, and use $\{\text{var}\}_{[m]}$ to denote the converged solution for m-th load increment.

4.8.1 Solution algorithm

1. Given the load increment number M, maximum iteration number Imax, also set both the initial load increment index m and initial iteration index i as 1.
2. For current load increment m, compute the stiffness matrix [K] based on current geometry (reference configuration) and set the displacement $\{U\}_{[m]}^{0} = \{0\}$. Set the second Piola-Kirchhoff stress $\{T\}^{(0)} = \{\sigma\}$, $\{\sigma\}$ is the Cauchy stress.
3. For iteration index i, we use Equation (52)-(54) to update the solutions.

$$\{F\}_{[m]}^{(i-1)} = K\{U\}_{[m]}^{(i-1)} - LEP^{(m)}\{U\}_{[m]}^{(i-1)} \tag{53}$$

$$[K]\{\Delta U\}^{\{i\}} = \{\Delta R\} - \{F\}_{[m]}^{(i-1)} \tag{54}$$

$$\{U\}_{[m]}^{(i)} = \{U\}_{[m]}^{(i-1)} + \{\Delta U\}^{(i)} \tag{55}$$

Where $\{\Delta R\}$ is the load increment and $LEP\{U\}_{[m]}^{(i-1)}$ are the nonlinear terms (the third-sixth terms of the Equation (47)).

4. The following convergence criterion is used to terminate the iteration for each load step:

$$\left\| \{\Delta R\} - \{F\}_{[m+1]}^{(i)} \right\|_2 \le \varepsilon_f \left\| \{\Delta R\} \right\|_2 \tag{56}$$

ε_f is a predefined tolerance, usually in the order of $10^{-4} \sim 10^{-7}$.

4.1. If the program converges in this iteration, update the geometry, the displacement field, and the second Piola-Kirchhoff stress for each Gaussian point (Section 4.8.2) and obtain the corresponding Cauchy stress by Equation (57). Go to step 5.

$$\sigma_{IJ} = (\delta_{IK} + \Delta u_{I,K})(T_{KL}^{0} + \Delta T_{KL})(\delta_{JL} + \Delta u_{J,L})/J \tag{57}$$

Where J is the determinant of the deformation gradient F which is defined as follows:

$$F = \begin{bmatrix} \dfrac{\partial x}{\partial X} & \dfrac{\partial x}{\partial Y} \\ \dfrac{\partial y}{\partial X} & \dfrac{\partial y}{\partial Y} \end{bmatrix} \tag{58}$$

4.2. Otherwise, increase the iteration index by one and check if i is greater than Imax. If i is greater than Imax, the program cannot converge for the preset Imax, exit program execution; otherwise, compute the stresses at all Gaussian points for all nodes and go to step 3.

5. Increase the load increment index m by one and check if m is greater than load increment number M. If Yes, exit the program; otherwise, go to step 2. The integrals for next load increment are to be performed over the deformed geometry of converged solution of current load increment (the current configuration in this load increment will be the reference configuration for the next load increment). This is the reason that it is called updated-Lagrangian formulation.

As shown in the previous section, when we calculate the internal force $\{F\}_{[m]}^{(i-1)}$, we need to calculate the $LEP\{U\}_{[m]}^{(i-1)}$ which are the nonlinear terms (the third-sixth terms of the Equation (47)). The integration is usually performed using Gaussian numerical integration techniques. Therefore, the stress state (along with the stress correction term) is computed at all Gaussian points in each iteration step. The following section gives the algorithm to compute the stress at each Gaussian point.

4.8.2 The procedure used to compute the stress at each Gaussian point

1. Assume the solution is converged for (m-1)th load increment, we transform the second Piola-Kirchhoff stresses which are basic stress measure obtained from solution to Cauchy stress. Cauchy stress obtained is then used as the initial value of the second Piola-Kirchhoff stress $\{T\}_{[m]}^{(0)}$ for the (m)th load increment. Given $\{\Delta u\}=^{(m+1)}\{U\}^{(i)}-^{m}\{U\}$, the strain increment $\{\Delta E\}$ and stress increment $\{\Delta T^e\}$ are initially predicted assuming elastic behavior using (59) and (60) shown below. $\{\Delta\varepsilon\}$ and $\{\Delta T'^e\}$ which is used to computed the fourth term of Equation (47) are predicted by equation (61) and (62). Set $\{T^{cor}\}_{[m]}$ and $\{T'^e\}_{[m]}$ zero. EPF is used to indicate the stress state.

$$\Delta E_{IJ} = \frac{1}{2}(\Delta u_{I,J} + \Delta u_{J,I} + \Delta u_{K,I}\Delta u_{K,J}) \tag{59}$$

$$\{\Delta T^e\} = [C^e]\{\Delta E\} \tag{60}$$

$$\Delta\varepsilon_{IJ} = \frac{1}{2}(\Delta u_{I,J} + \Delta u_{J,I}) \tag{61}$$

$$\{\Delta T'^{e}\} = [C^{e}]\{\Delta\varepsilon\} \tag{62}$$

2. Determine the loading state

2.1 If EFP=1, the Gaussian point is in an elastoplastic state in previous load step. Compute the loading criterion function L using Equation (18).

If L>0, r=0, plastic loading; if L<=0, let r=1, EPF=0, compute the yield function after the trial stress increment is applied

$$f(\{T\}_{[m]} + \{\Delta T^{e}\},\ \kappa_{[m]})$$

if $f \le 0$, r=1, the point remains in the elastic state; if $f > 0$, EPF=1, the point enters into elastoplastic state determine r such that (63) shown between holds.

$$f(\{T\}_{[m]} + \{\Delta T^{e}\},\ \kappa_{[m]}) = 0 \tag{63}$$

Update the stress at this point by

$$\{T\} = \{T\}_{[m]} + r\{\Delta T^{e}\} \tag{64}$$

2.2 If EPF=0, the Gaussian point is in the elastic state in the previous load step. Compute the yield function after the trial stress increment is applied

$$f(\{T\}_{[m]} + \{\Delta T^{e}\},\ \kappa_{[m]})$$

if $f \le 0$, r=1, the point remains in the elastic state; if $f > 0$, EPF=1, the point enters into elastoplastic state. Determine r such that (63) holds. Update the stress at this Gaussian point using Equation (64).

3. Compute the sub-increment of strain $\{\Delta\tilde{E}\}$:

$$\{\Delta\tilde{E}\} = \frac{(1-r)\{\Delta E\}}{N} \tag{65}$$

where N is a integer. Integrate numerically to compute sub-increment of stress $\{\Delta\tilde{T}_{ij}\}$ with n looping from 1 to N, $\{\Delta T\}^{(0)} = \{\Delta T^{cor}\} = 0$:

$$\begin{aligned}\Delta\tilde{T}_{ij} &= 2\mu\Delta\tilde{E}_{ij} + \lambda\Delta\tilde{E}_{kk}\delta_{ij} - \frac{9\mu^2}{3\mu + H_p}\frac{S_{kl}\Delta E_{kk}S_{ij}}{\sigma_e^2} \\ &+ [-\frac{1}{2}(\delta_{IK}\sigma_{JL} + \delta_{JK}\sigma_{IL} + \delta_{IL}\sigma_{JK} + \delta_{JL}\sigma_{IK}) + \sigma_{IJ}\delta_{KL}]\Delta\tilde{E}_{KL}\end{aligned} \tag{66}$$

$$\{\Delta T\}^{(n)} = \{\Delta T\}^{(n-1)} + \{\Delta \widetilde{T}\} \tag{67}$$

4. Update the variables:

$$\{\Delta T^{cor}\} = \{\Delta T\}^{(N)} + r\{\Delta T^{e}\} - \{\Delta T'^{e}\} \tag{68}$$

$$\{T^{cor}\}^{(i)}_{[m]} = \{T^{cor}\}_{[m]} + \{\Delta T^{cor}\} \tag{69}$$

$$\{T\}^{(i)}_{[m]} = \{T\}^{0}_{[m]} + \{\Delta T\}^{(N)} + r\{\Delta T^{e}\} \tag{70}$$

$$\{T'^{e}\}^{(i)}_{[m]} = \{T'^{e}\}_{[m]} + \{\Delta T'^{e}\} \tag{71}$$

4.8.3 The computation of nonlinear terms

In order to calculate the third term in Equation (47), we also need to compute $\widetilde{u}^{*}_{IJ,K}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)$, which is the derivative of $\widetilde{u}^{*}_{IJ}(\mathbf{X},\mathbf{Y}^{<a>})$ with respect to X_K. We use (69) to calculate $\{T^{cor}\}$ for each Gaussian point, and compute the third term in Equation (47) according. We use (71) to calculate $\{T'^{e}\}$ which is equal to $C^{e}_{LKMN}U_{M,N}$ and compute the fourth term of Equation (47).

As for the fifth and sixth terms of Equation (47), T^{0}_{LK} is the initial value of the second Piola-Kirchhoff stress of this load increment. ΔT_{LK} is the increment of the second Piola-Kirchhoff stress of the current iteration with respect to T^{0}_{LK}. These two terms can be computed easily.

4.9 Numerical Examples

This section presents the numerical solutions to several examples using the meshless integral method for elastoplasticity with large deformation. The tests include the uniaxial tension tests, the shear tests and the rotation tests. For the elasticity with large deformation cases, we can use hand calculation solutions for comparison. For the elastoplasticity with large deformation, we use the finite element results obtained using commercial software, ANSYS, as the bases for comparison.

4.9.1 Uniaxial tension tests

The patch test is a 1×1 m^2 square plate shown in Figure 4.7. The material is AISI 1020 steel and the material properties are E=203000 MPa, $\nu = 0.3$ with plane strain condition. For the elastoplastic case, yield stress is equal to 260 MPA, and the elastoplastic tangent modulus is 1000 MPA. Three meshless models, which are shown in Figure 4.7, were simulated. The left edge is constrained from moving in x1 direction but is traction free in x2 direction. The bottom edge is constrained from moving in x2 direction but is traction free in x1 direction. The prescribed displacement Ux (0.1, 0.25, and 0.5) in x1 is applied to right edge which is traction free in x2 direction. The nature boundary conditions (free traction) are prescribed on the top edge. The spline weight function is used.

For case 1 in which Ux=0.1, we first perform the convergence test for elastic case as we increase the number of load increments. Figure 4.8a-4.8c show that the meshless results (σ_{11}, σ_{33}, and displacement of upper edge b) converge for elastic case. The msehless results are identical with the hand calculation results and we list the hand calculation results and meshless results for one load increment and two load increments in Table 4.2 and Table 4.3 respectively. The meshless results for 12 load increments are σ_{11}=21428, σ_{33}=6430, and b= -0.040473. With 12 load increments, the corresponding results of FEM are σ_{11}=21261, σ_{33}=6378, and b=-0.04027. Figure 4.9 and Figure 4.10 show FEM results versus meshless results for uniaxial tension tests of elastic case and indicate good agreement between meshless results and FEM results.

For the elastoplasticity when Ux=0.1, there are no hand calculation results so we compare the meshless results with FEM results. It is shown that both meshless results and FEM results converge as the number of load increments increase. The converged values for FEM and meshless method are very close to each other. The meshless results for 12 load increments are σ_{11}=426, σ_{33}=212, and b= -0.090088. The corresponding results of FEM with 12 load increments are σ_{11}=426, σ_{33}=212, and b= -0.089776. Figure 4.11a-4.11c show that the FEM results and meshless results for σ_{11}, σ_{33}, and displacement of upper edge b converge for elastoplastic case. Figure 4.12 and Figure 4.13 show FEM results versus meshless results for uniaxial tension tests of elastoplasric case and indicate that meshless results follow closely FEM results. It is noteworthy that this meshless method is based on Green-Naghdi's theory, while

FEM is based on E.H.Lee's theory [2]. The differences between meshless results and FEM results show that large deformation elastoplasticity theories may have some influence on the numerical results.

4.9.2 The shear tests

The shear patch test is a 1×1 m^2 square plate shown in Figure 4.14. The material is AISI 1020 steel and the material properties are E=203000 MPa, $\nu = 0.3$ with plane strain condition. For the elastoplastic cases, yield stress is equal to 260 MPA, and the elastoplastic tangent modulus is 1000 MPA. Three meshless models, which are shown in Figure 4.14, were simulated. The spline weight function is used in the shear tests. We simulate the tests when shear strain $\gamma = 0.1$ and $\gamma = 0.5$. The motion of the square is described by $x = X + \gamma Y$ and $y = Y$. For all three models, prescribed displacements are applied to all edges of the plate.

Analytical results in terms of the Cauchy stress using Jaumann rate for this problem are given as [36]:

$$\sigma_{11} = -\sigma_{22} = \mu(1 - \cos\gamma) \tag{71}$$

$$\sigma_{12} = \mu \sin\gamma \tag{72}$$

For $\gamma = 0.1$ with elasticity, the L_2-norm for the 9 node, 25 node regular, and 25 node irregular modes are 3.2×10^{-16}, 2.6×10^{-18}, and 3.6×10^{-17} respectively. The stresses for all three cases are $\sigma_{11} = 390$, $\sigma_{22} = -391.48$, $\sigma_{12} = 7794.88$ which are pratically identical with the analytical solution using Jaumann rate: $\sigma_{11} = 390$, $\sigma_{22} = -390$, $\sigma_{12} = 7794.68$.

For $\gamma = 0.5$ with elasticity, the L_2-norm for the 9 node, 25 node regular, and 25 node irregular modes are 6.2×10^{-16}, 5.5×10^{-18}, and 4.7×10^{-16} respectively. The stresses for all three cases are $\sigma_{11} = 9560$, $\sigma_{22} = -9555$, $\sigma_{12} = 37431$ which are pratically identical with the analytical solution using Jaumann rate: $\sigma_{11} = 9557$, $\sigma_{22} = -9557$, $\sigma_{12} = 37432$.

Figure 4.15 shows analytical versus meshless results for pure shear simulation of elastic case. It is indicated that meshless results match analytical results excellently.

For the elastoplastic cases, we compare the meshless results with FEM results obtained using ANSYS.

For $\gamma = 0.1$ with elastoplasticity, the L_2-norm for the 9 node, 25 node regular, and 25 node irregular modes are 3.7×10^{-13}, 4.6×10^{-13}, and 6.4×10^{-14} respectively. The shear stress

for all three cases is $\sigma_{12} = 182.32$ which are in excellent agreement with the FEM solution of $\sigma_{12} = 182.60$.

For $\gamma = 0.5$ with elastoplasticity, the L_2-norm for the 9 node, 25 node regular, and 25 node irregular modes are 5.2×10^{-13}, 1.8×10^{-13}, and 9.4×10^{-11} respectively. The stresses for all three cases are $\sigma_{12} = 316.52$ which are again in good agreement with the FEM solution of $\sigma_{12} = 316.89$.

Figure 4.16 shows FEM results versus meshless results for pure shear simulation of elastoplastic case. We can see that meshless results are in excellent agreement with FEM results.

4.9.3 The rigid body rotation tests

This rigid body rotation patch test is a 1×1 m^2 square plate shown in Figure 4.17. Figure 4.17a is the original configuration, while Figure 4.17b is the current configuration with rotation angle θ. The material is AISI 1020 steel and the material properties are E=203000 MPa, $\nu = 0.3$ with plane strain condition. Three meshless models (9 regular nodes, 25 regular nodes, and 25 irregular nodes), were simulated. The spline weight function is used in the rotation tests. We simulate the tests with $\theta = \frac{\pi}{12}$, $\theta = \frac{\pi}{6}$, $\theta = \frac{\pi}{4}$, $\theta = \frac{\pi}{3}$, $\theta = \frac{5\pi}{12}$, and $\theta = \frac{\pi}{2}$. For all three models, prescribed displacements are applied to all edges of the plate. Initial stresses are $\sigma_{11}^0 = 20$, $\sigma_{22}^0 = 0$, $\sigma_{12}^0 = 0$.

For $\theta = \frac{\pi}{12}$, $\sigma_{11} = 18.66$, $\sigma_{22} = 1.3397$, $\sigma_{12} = 5$; for $\theta = \frac{\pi}{6}$, $\sigma_{11} = 15$, $\sigma_{22} = 5$, $\sigma_{12} = 8.66$; for $\theta = \frac{\pi}{4}$, $\sigma_{11} = 10$, $\sigma_{22} = 10$, $\sigma_{12} = 10$; for $\theta = \frac{\pi}{3}$, $\sigma_{11} = 5$, $\sigma_{22} = 15$, $\sigma_{12} = 8.66$; for $\theta = \frac{5\pi}{12}$, $\sigma_{11} = 1.3397$, $\sigma_{22} = 18.66$, $\sigma_{12} = 5$; and for $\theta = \frac{\pi}{2}$, $\sigma_{11} = 0$, $\sigma_{22} = 20$, $\sigma_{12} = 0$. All these results are identical with the analytical results given by the following equation [36]:

$$\boldsymbol{\sigma} = \sigma_{11}^0 \begin{bmatrix} \cos^2\theta & \frac{1}{2}\sin 2\theta \\ \frac{1}{2}\sin 2\theta & \sin^2\theta \end{bmatrix} \tag{73}$$

Figure 4.18-4.20 show analytical versus meshless results (σ_{11}, σ_{12}, σ_{11}) for rotation tests for different rotation angle theta. We can see that the meshless results are identical with the analytical results.

For the three meshless models with different nodal densities, we obtained the identical results.

4.10 Concluding Remarks

In this paper, the large deformation elastoplastic meshless integral method based on regularized boundary integral equation [1] is presented. The updated Lagrangian governing integral equation is obtained from the weak form of elastoplasticity over a local sub-domain based on Green-Naghdi's theory. Green-Naghdi's theory starts with the decomposition of the Green-Lagrange strain into the elastic part and plastic part and considers a J_2 elastoplastic constitutive relation that relates the Green-Lagrange strain to second order Piola-Kirchhoff stress. The collocation method is employed to enforce the essential boundary conditions exactly, which is simple and computationally efficient. The natural boundary conditions are incorporated in the system governing equation and require no special handling. Numerical results show that this method is accurate and robust.

4.11 References

[1] Anthony Bodin, Jianfeng Ma, X.J. Xin, and Prakash Krishnaswami. A meshless integral method based on regularized boundary integral equation. Computer Methods in Applied Mechanics and Engineering. 2006; 195: 6258-6286.

[2] Chiou, J. H., Lee, J. D., and Erdman, A. G. Comparison between two theories of plasticity. Computers & Structures. 1986: 24(1): 23-37.

[3] Chiou, J. H., Lee, J. D., and Erdman, A. G. Development of a three-dimensional finite element program for large strain elastic-plastic solids. Computers & Structures. 1990: 36 (4): 631-645.

[4] Lee, J. D. A large-strain elastic-plastic finite element analysis of rolling process. Computer Methods in Applied Mechanics and Engineering. 1998; 161: 315-347.

[5] Hu, P. Finite-element numerical analysis of sheet metal under unaxial tension with a new yield criterion. Journal of Materials Processing Technology. 1992; 31: 245-253.

[6] Belytschko, T., Krysl, P., and Krongauz, Y. A three-dimensional explicit element-free Galerkin method. International Journal for Numerical methods in Fluids. 1997; 24: 1253-1270.

[7] Rossi, R. and Alves, M. K. On the analysis of an EFG method under large deformations and volumetric locking. Computational Mechanics. 2007; 39: 381-399.

[8] Chen, Y. P., Eskandarian, A., Oskard, M., and Lee, J. D. Meshless analysis of high-speed impact. Theoretical and Applied Fracture Mechanics. 2005; 44: 201-207.

[9] Eskandarian, A., Chen, Y. P., Oskard, M., and Lee, J. D. Meshless analysis of fracture. Plasticity and impact. Proceedings of IMECE'03. 2003 ASME International Mechanical Engineering Congress. Washington, D.C., November 15-21, 2003.

[10] Jianfeng Ma, X.J. Xin, and Prakash Krishnaswami. An elastoplastic meshless integral method based on regularized boundary integral equation. Computer Methods in Applied Mechanics and Engineering. (to be submitted)

[11] Stroud, A.H. and Secrest, D. *Gaussian Quadrature Formulas*. Prentice-Hall, 1966.

[12] Lu, YY, Belytschko T, and Gu, L, A new implementation of the element free Galerkin method, Computer Methods in Applied Mechanics and Engineering, 1994, 113, pp.397-414

[13] Belytschko, T., Lu, Y.Y. and Gu, L. Element free Galerkin method. International Journal for Numerical Methods in Engineering. 1994; 37: 229-256.

[14] Gavete, L., Benito, J.J., Falcon S. and Ruiz A. Implementation of essential boundary conditions in a meshless method. *Commun. Numer. Meth. Engng.* 2000; 16: 409-421

[15] Zhu, T. and Atluri, S.N. A modified collocation method and a penalty foumulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput. Mech. 1998; 21(3): 211-222.

[16] Arnold DN, Brezzi F, Cockburn B, Marini LD, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIMA J. Numer. Anal., 2002, 39, 1749-1779

[17] Hegen D., Element-free Galerkin methods in combination with finite element approaches, Computer Methods in Applied Mechanics and Engineering, 1996, 19, 120-135

[18] Gosz, J. and Liu, W.K. Admissible approximations for essential boundary conditions in the reproducing kernel particle method. Comput. Mech. 1996; 19(2):120-135

[19] Gunther, F. C. and Liu, W. K. Implementation of boundary conditions for meshless methods. Computer Methods in Applied Mechanics and Engineering 1998; 163: 205-230

[20] Duarte CAM and Oden JT, An h-p adaptive method using clouds, Comput. Methods Appl. Mech. Engrg, 1996, 139, 237-262.

[21] Lu YY, Belytschko T, Tabbara M. Element-free Galerkin method for wave propagation and dynamic fracture, Computer Methods in Applied Mechanics and Engineering, 1995, 126, pp.131-153.

[22] Zhang X, Liu X, Lu MW and Chen Y, Imposition of essential boundary conditions by displacement constraint equations in meshless methods, Communications in Numerical Methods in Engineering, 2001, 17, 165-178.

[23] Wagner, GJ and Liu WK, Application of essential boundary conditions in mesh-free methods: a corrected collocation method, International Journal for Numerical Methods in Engineering, 2000, 47, 1367-1379

[24] Zhu, T. and Atluri, S.N. A modified collocation method and a penalty foumulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput. Mech. 1998; 21(3): 211-222.

[25] Wu CC and Plesha ME, Essential boundary condition enforcement in meshless methods: boundary flux collocation method, Int. J. Numer. Meth. Engng, 2002, 53, 499-514.

[26] Zhu T., Zhang J-D., and Atluri S.N. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Computational Mechanics. 1998; 21: 223-235.

[27] Atluri S.N., Sladeck J., Sladeck V., and Zhu T. The local boundary integral equation (LBIE) and its meshless implementation for linear elasticity. Computational Mechanics. 2000; 25: 180-198.

[28] Sladek, J., Sladeck, V., and Van Keer, R. Meshless local boundary integral equation for 2D elastodynamic problems. International Journal for Numerical Methods in Engineering. 2003: 235-249.

[29] Long, S. and Zhang. Q. Analysis of thin plates by the local boundary integral equation (LBIE) method. Engineering Analysis with Boundary Elements. 2002; 26: 707-718.

[30] Belytschko, T., Lu, Y.Y. and Gu, L. Element free Galerkin method. International Journal for Numerical Methods in Engineering. 1994; 37: 229-256.

[31] Krysl P, Belytschko T., Analysis of thin plates by the element-free Galerkin method, Computational Mechanics, 1998, 21, 211-222.

[32] Liu, W.K. and Chen. Y. Wavelet and multiple scale reproducing kernel methods. International Journal for Numerical Methods in Fluids. 1995; 21: 901-931.

[33] Aluru, N.R. A reproducing kernel particle method for meshless analysis of microelectromechanical systems. Computational Mechanics. 1999; 23: 324-338

[34] Chen, J.S., Pan, C., and Wu, C.T. A Lagrangian reproducing kernel particle method for metal forming analysis. Computational Mechanics. 1998; 22: 289-338.

[35] Chen, J.S., Pan, C., Wu, C.T., and Liu, W.K. Reproducing kernel particle methods for large deformation analysis of non-linear structures. Comput. Methods Appl. Mech. Engrg. 1996; 139 (1-4): 195-227.

[36] Belytschko, T., Liu, W. K, and Moran, B. Nonlinear Finite Elements for Continua and Structures. John Wiley & Sons, LTD. 2000. West Sussex, England.

Figure 4.1 Undeformed (initial or reference) and deformed (current) configuration

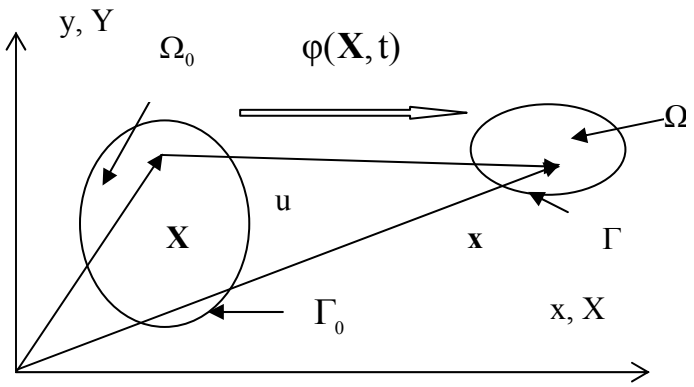


Figure 4.2 Schematic diagram showing the sub-domain for an interior or a boundary node $\mathbf{Y}^{<a>}$.

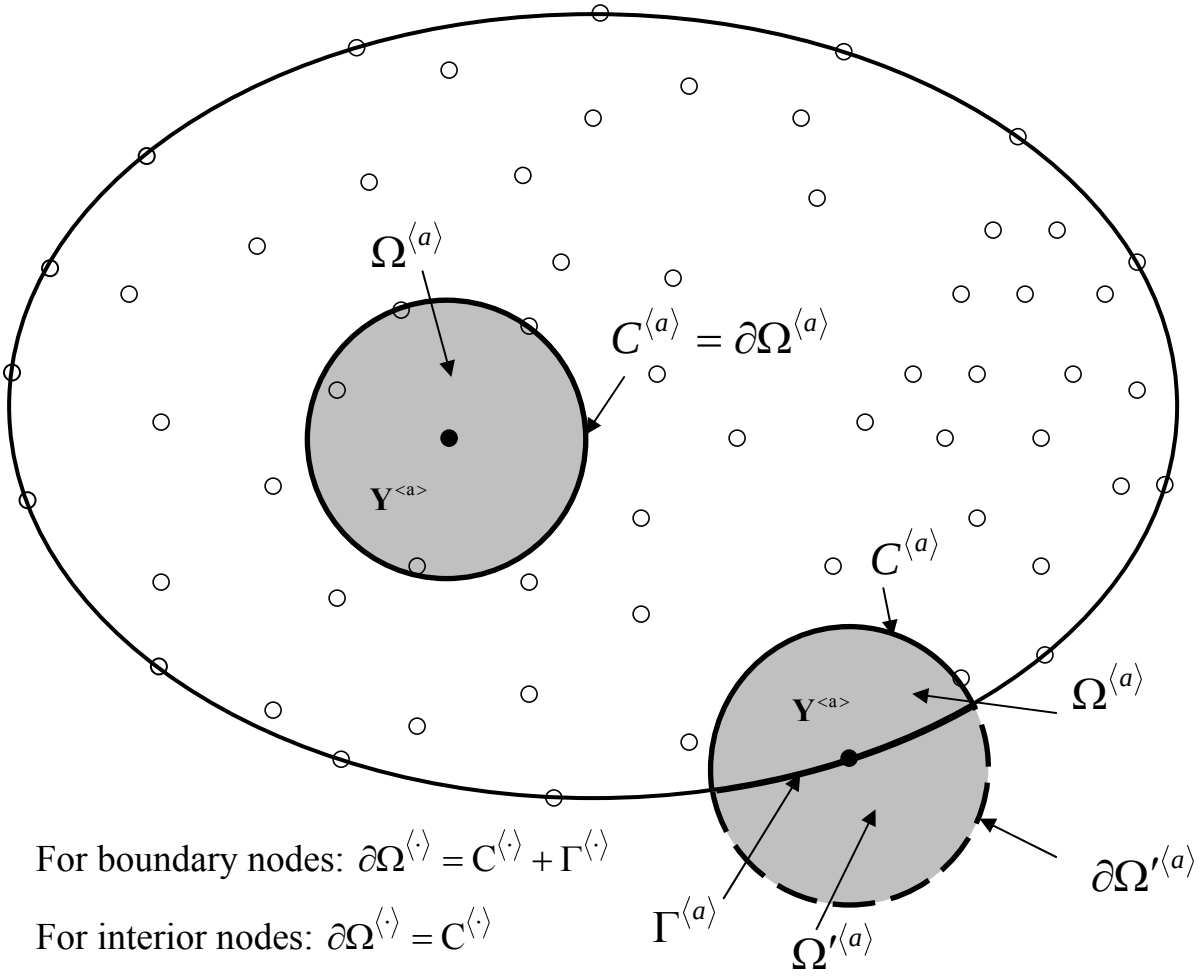
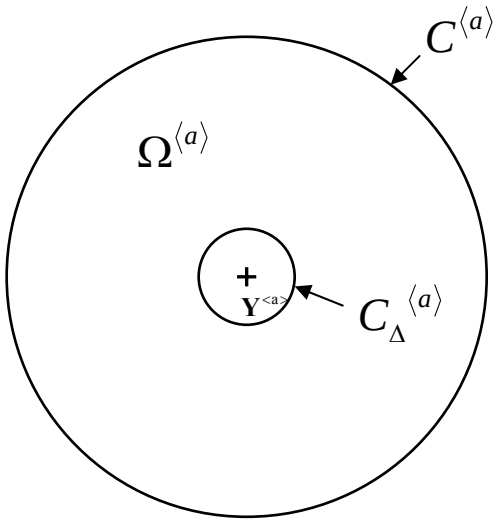


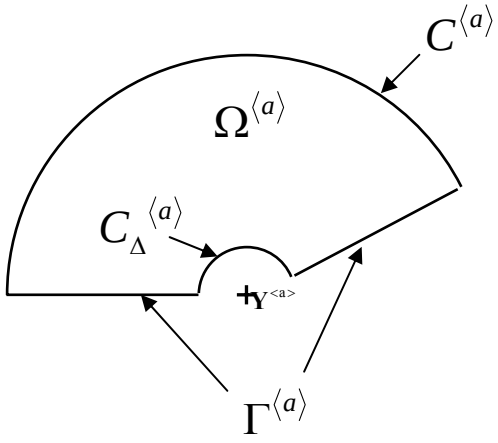
Figure 4.3 Exclusion of a tiny sphere Ω_Δ of radius Δ centered a node for removing the strong singularity



When $\mathbf{Y}^{<a>}$ is an interior node

$$\partial\Omega^{\langle a \rangle} = C^{\langle a \rangle} + C_\Delta{}^{\langle a \rangle} + \Gamma^{\langle a \rangle}$$

with $\Gamma^{\langle a \rangle} = \varnothing$



When $\mathbf{Y}^{<a>}$ is a boundary node

$$\partial\Omega^{\langle a \rangle} = C^{\langle a \rangle} + C_\Delta{}^{\langle a \rangle} + \Gamma^{\langle a \rangle}$$

Figure 4.4 Schematic diagram showing the internal boundary angle $\theta = \theta_2 - \theta_1$ at node $\mathbf{Y}^{<a>}$ on the boundary

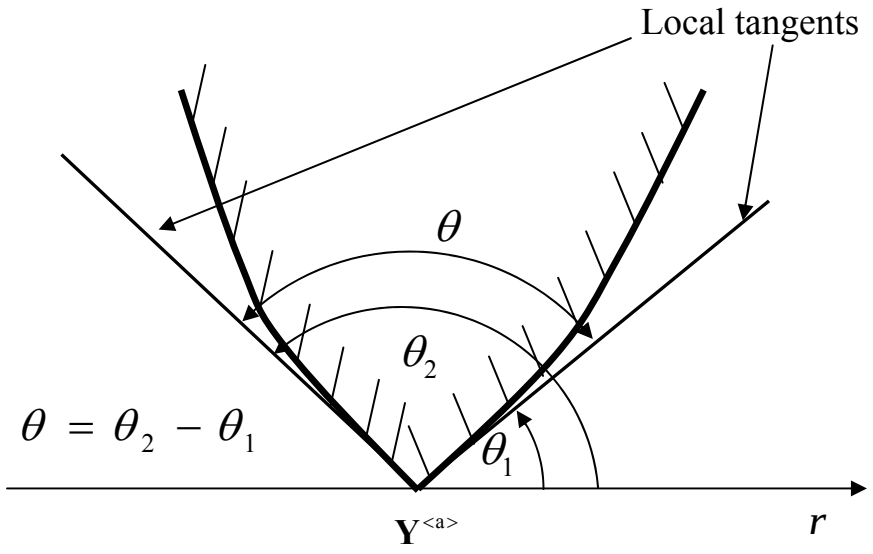


Figure 4.5 Schematic diagram illustrating the meaning of local sub-domain and support domain for node $\mathbf{Y}^{<a>}$ and node $\mathbf{Y}^{<b>}$

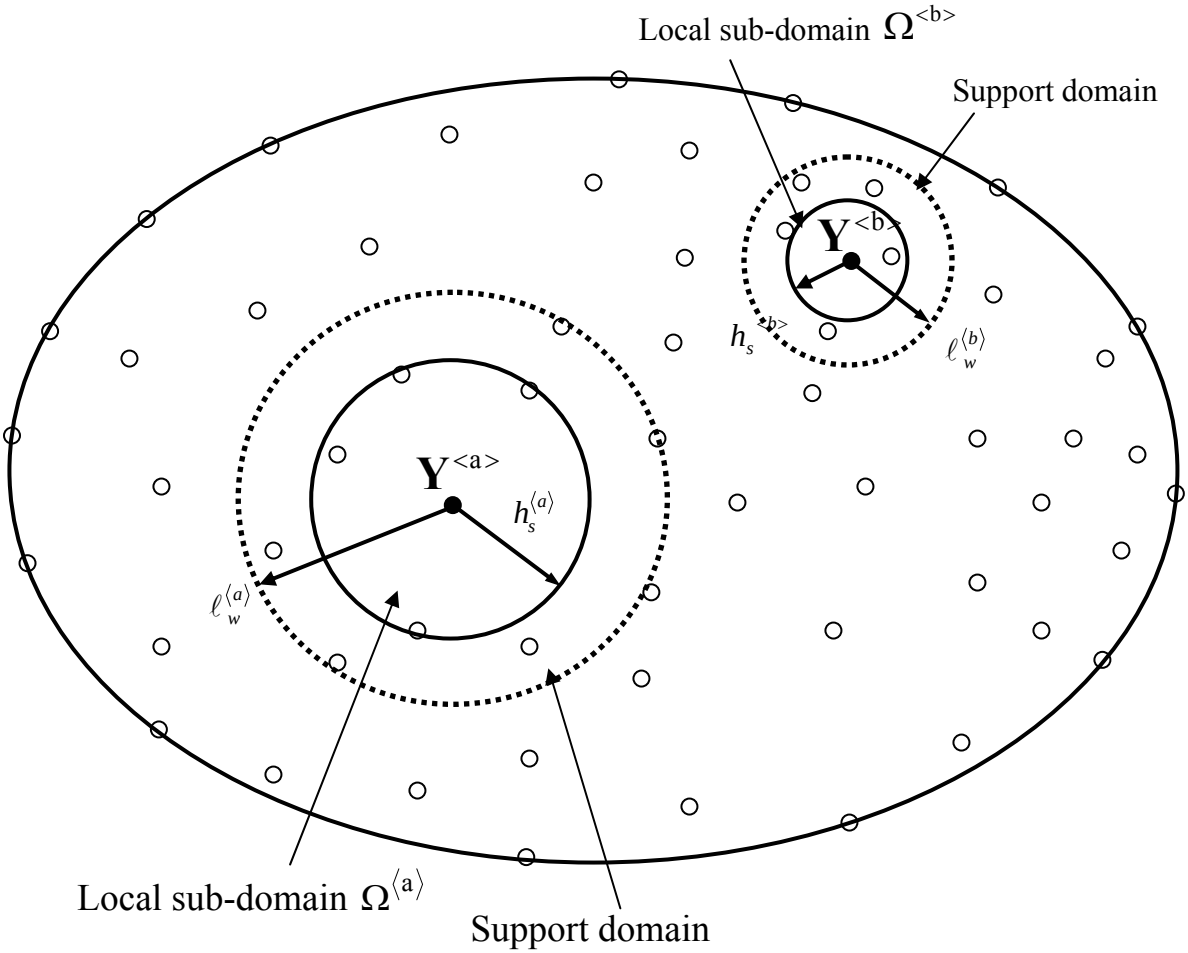


Figure 4.6 Schematic diagram showing the domain of influence for node $\mathbf{Y}^{<a>}$ and the domain of definition for a point $\mathbf{X}$

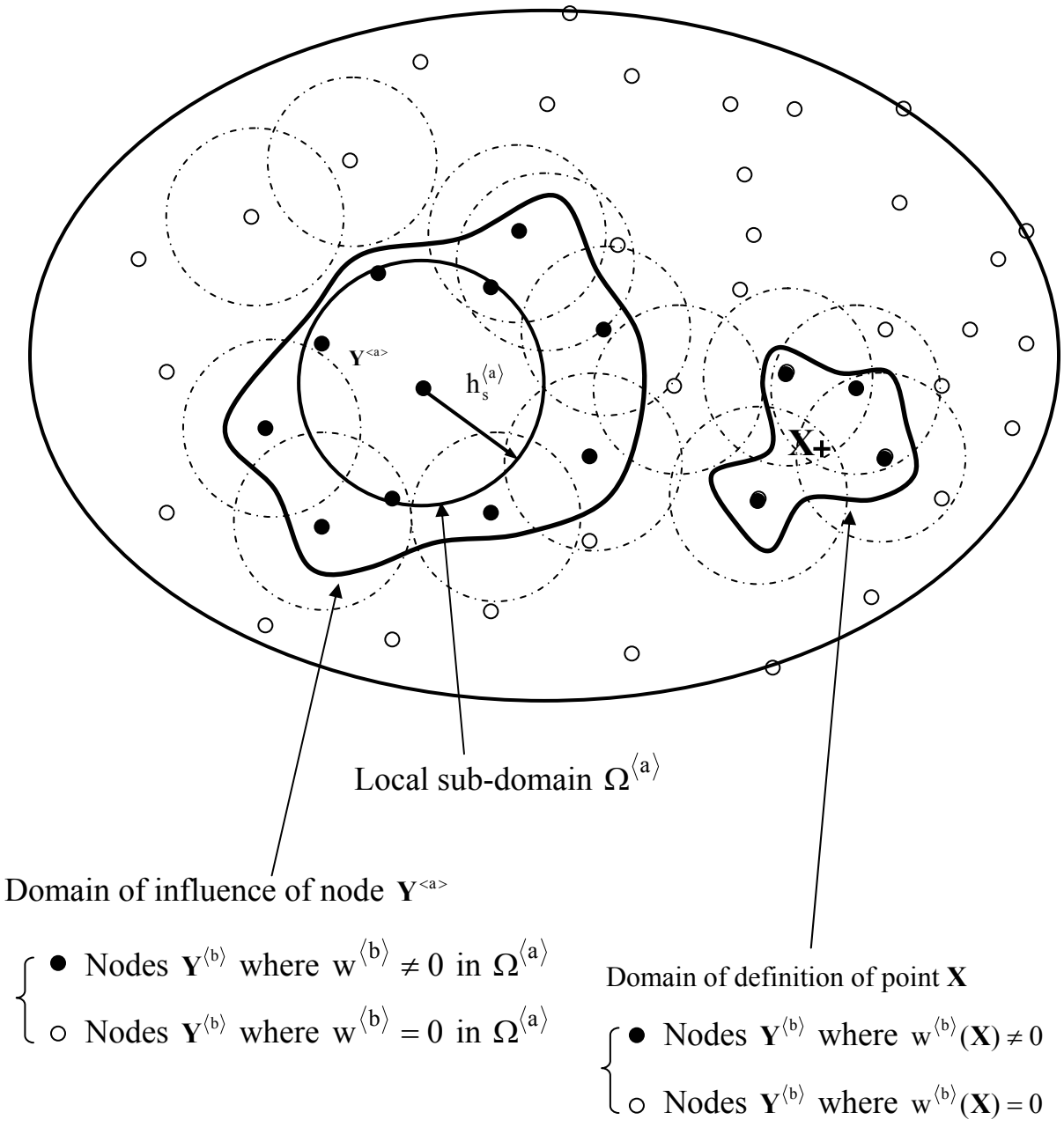
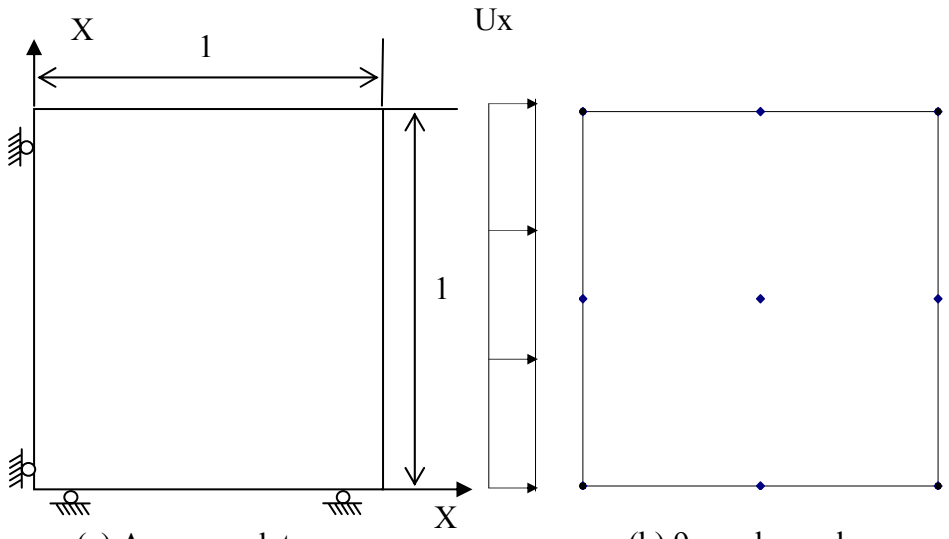
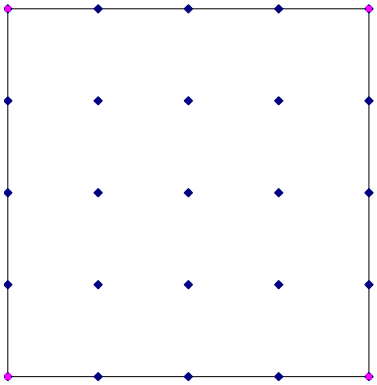


Figure 4.7 (a) A square plate for the patch test. (b) Meshless model with 9 regular nodes. (c) Meshless model with 25 regular nodes. (d) Meshless model with 25 irregular nodes.

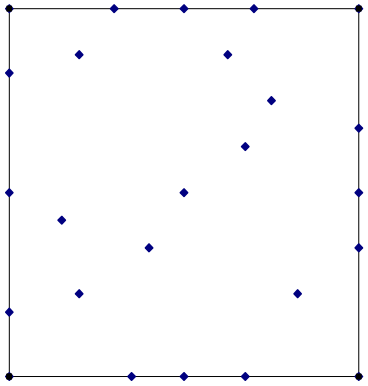


(a) A square plate

(b) 9 regular nodes

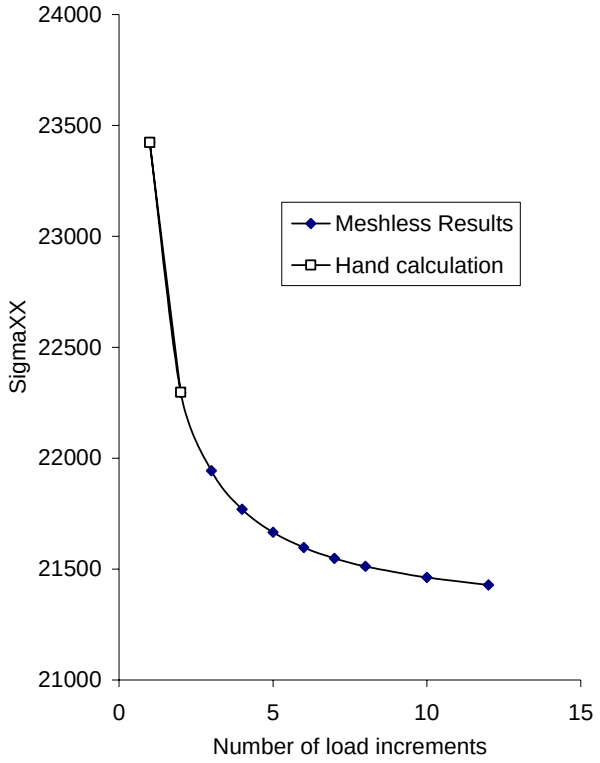


(c) 25 regular nodes

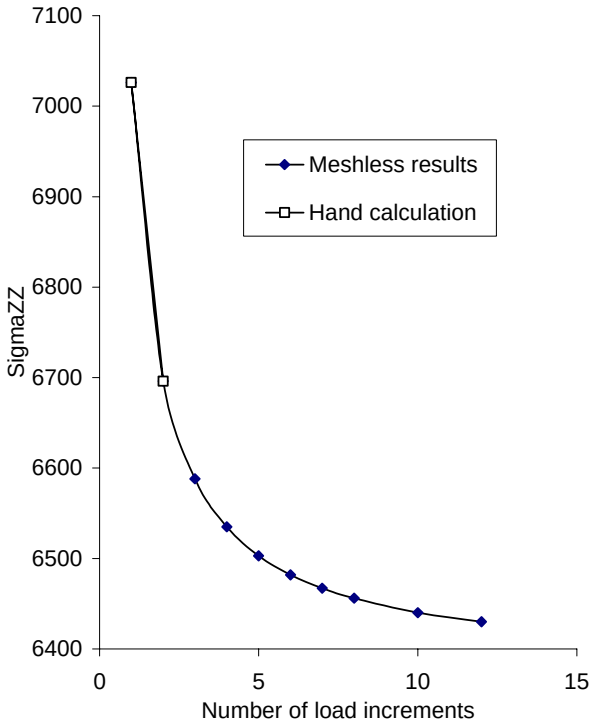


(d) 25 irregular nodes

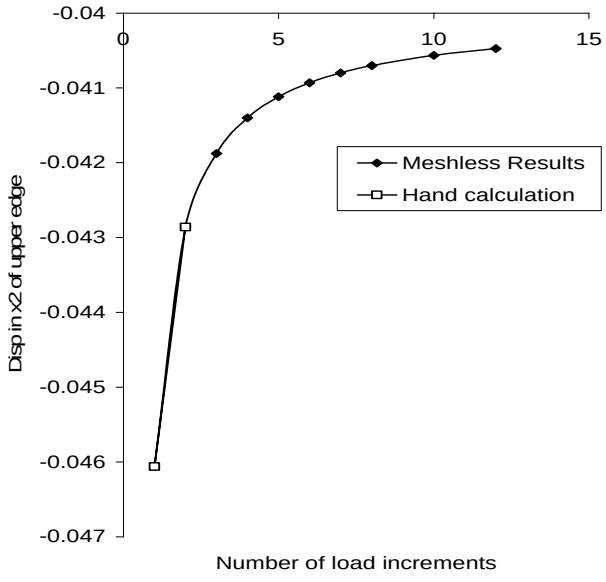
Figure 4.8 The convergence test of meshless method for σ_{11}, σ_{33}, and displacement of upper edge (elastic case)



(a)



(b)



(c)

Figure 4.9 FEM results versus meshless results σ_{11} for uniaxial tension simulation for elastic case

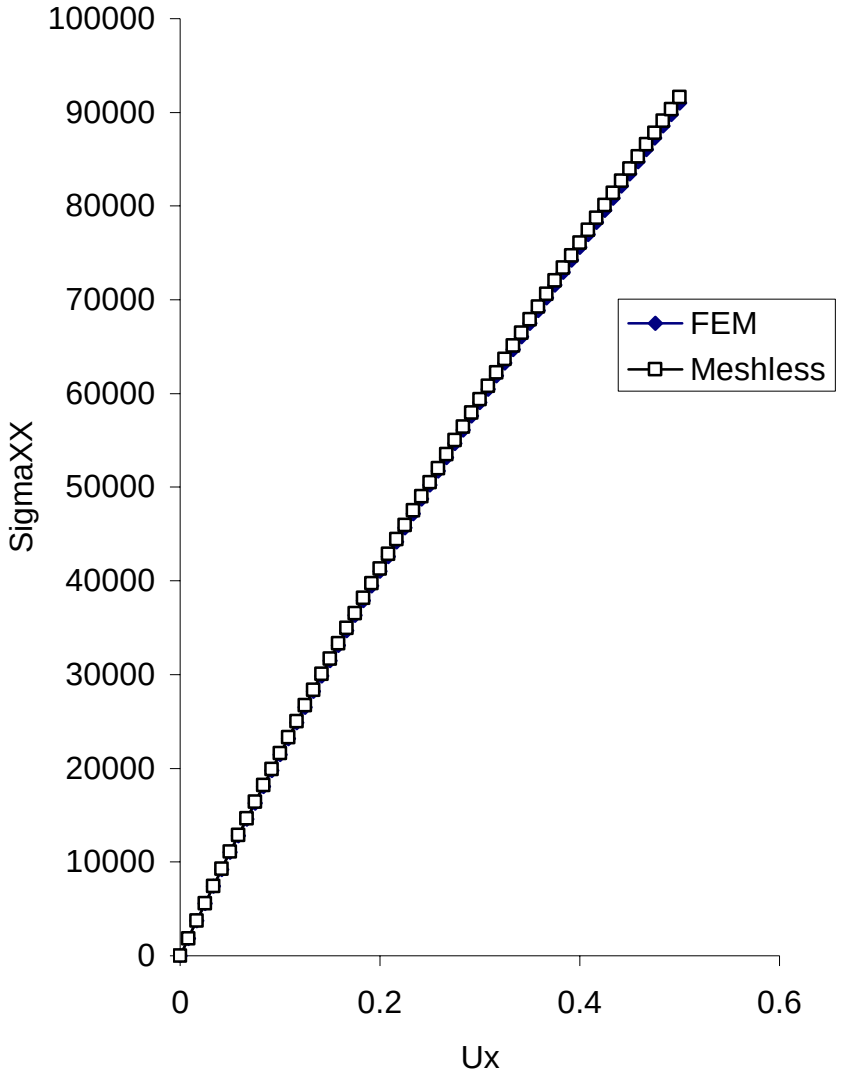


Figure 4.10 FEM results versus meshless results σ_{33} for uniaxial tension simulation for elastic case

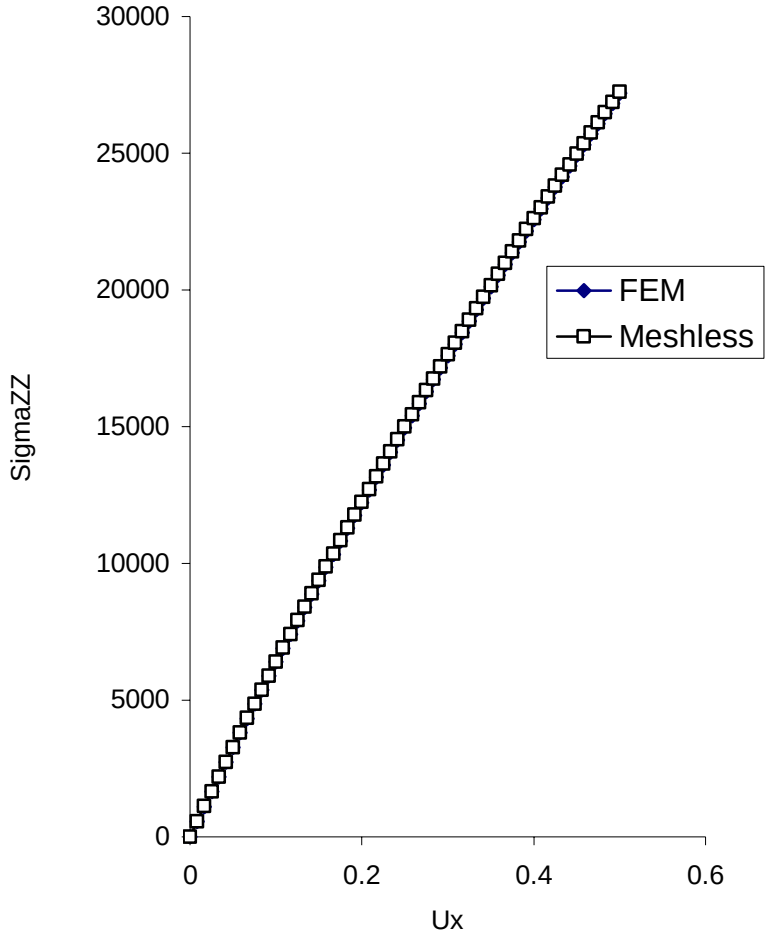
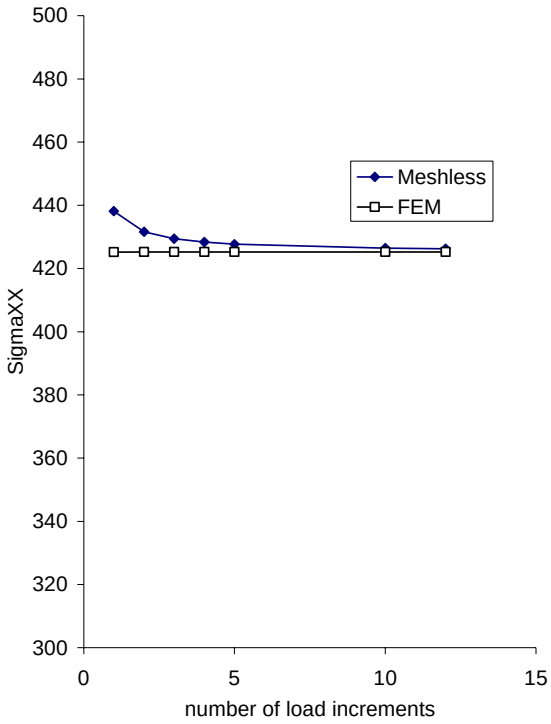
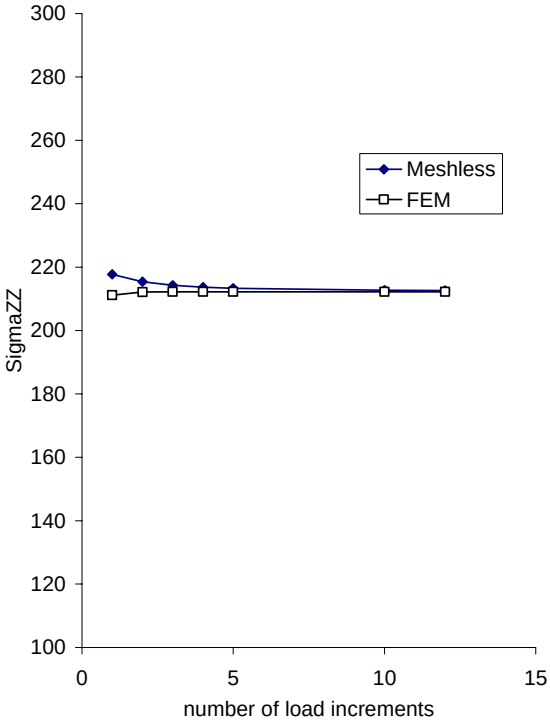


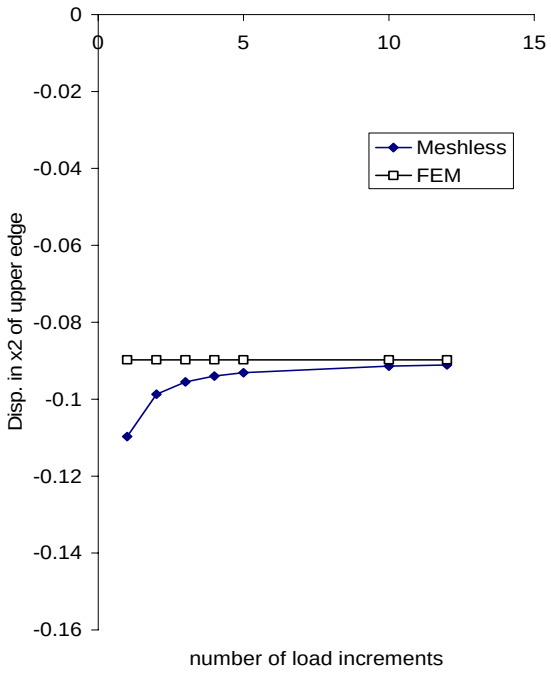
Figure 4.11 The convergence test of FEM results and meshless results for σ_{11}, σ_{33}, and displacement of upper edge (elastoplastic case)



(a)



(b)



(c)

Figure 4.12 FEM results versus meshless results σ_{11} for uniaxial tension simulation for elastoplastic case

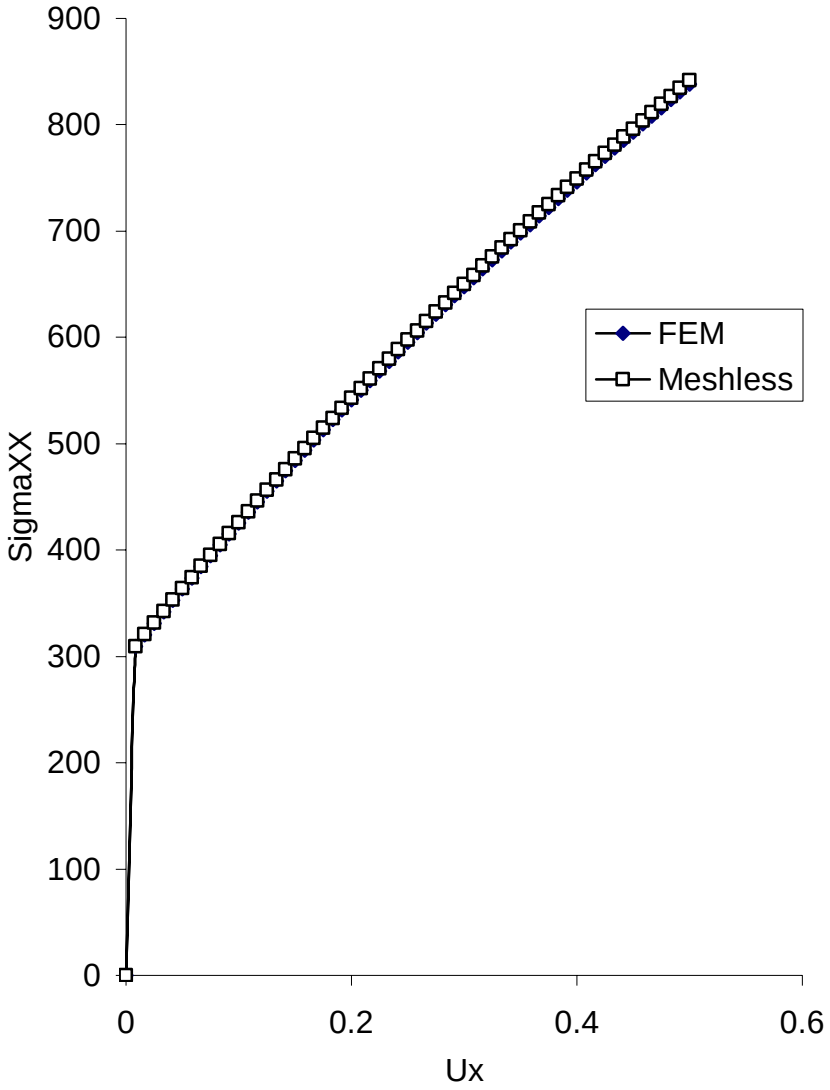


Figure 4.13 FEM results versus meshless results σ_{33} for uniaxial tension simulation for elastoplastic case

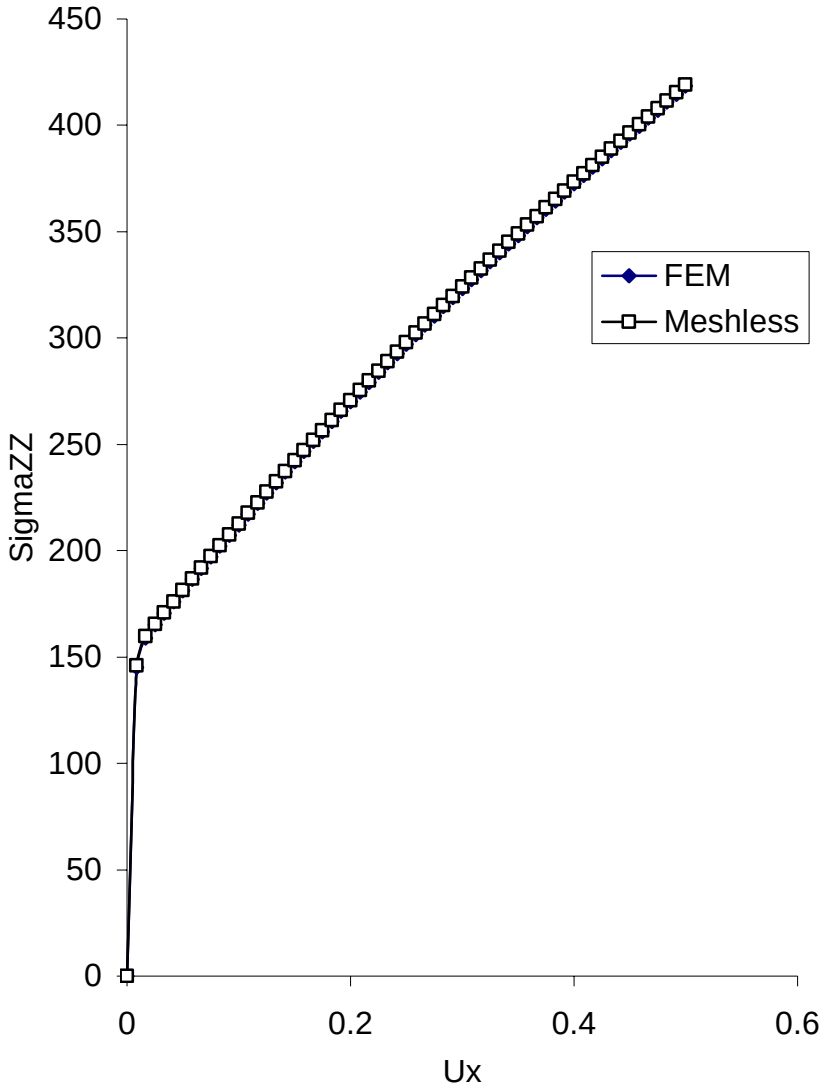
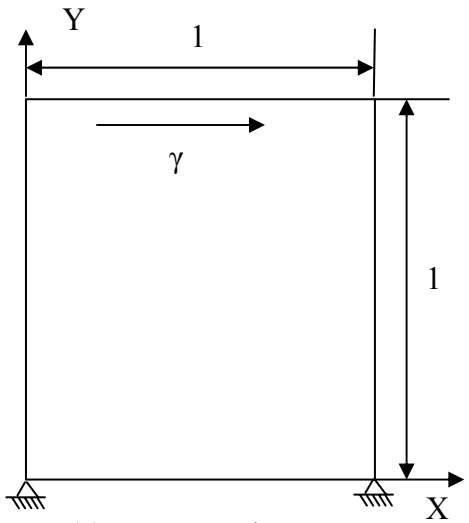
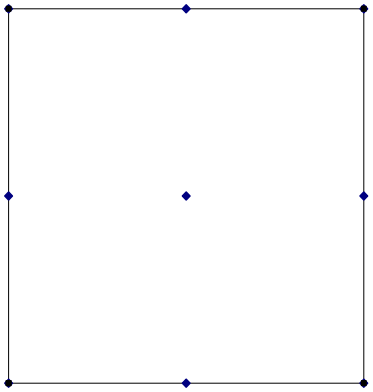


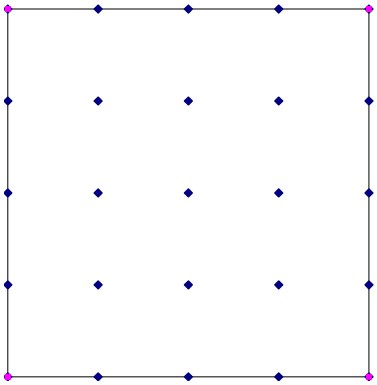
Figure 4.14 (a) A square plate for the shear tests. (b) Meshless model with 9 regular nodes. (c) Meshless model with 25 regular nodes. (d) Meshless model with 25 irregular nodes



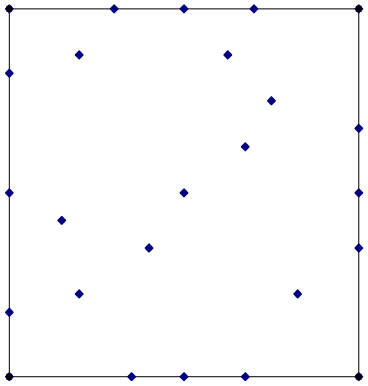
(a) A square plate



(b) 9 regular nodes



(c) 25 regular nodes



(d) 25 irregular nodes

Figure 4.15 Analytical versus meshless results for pure shear simulation for elastic case

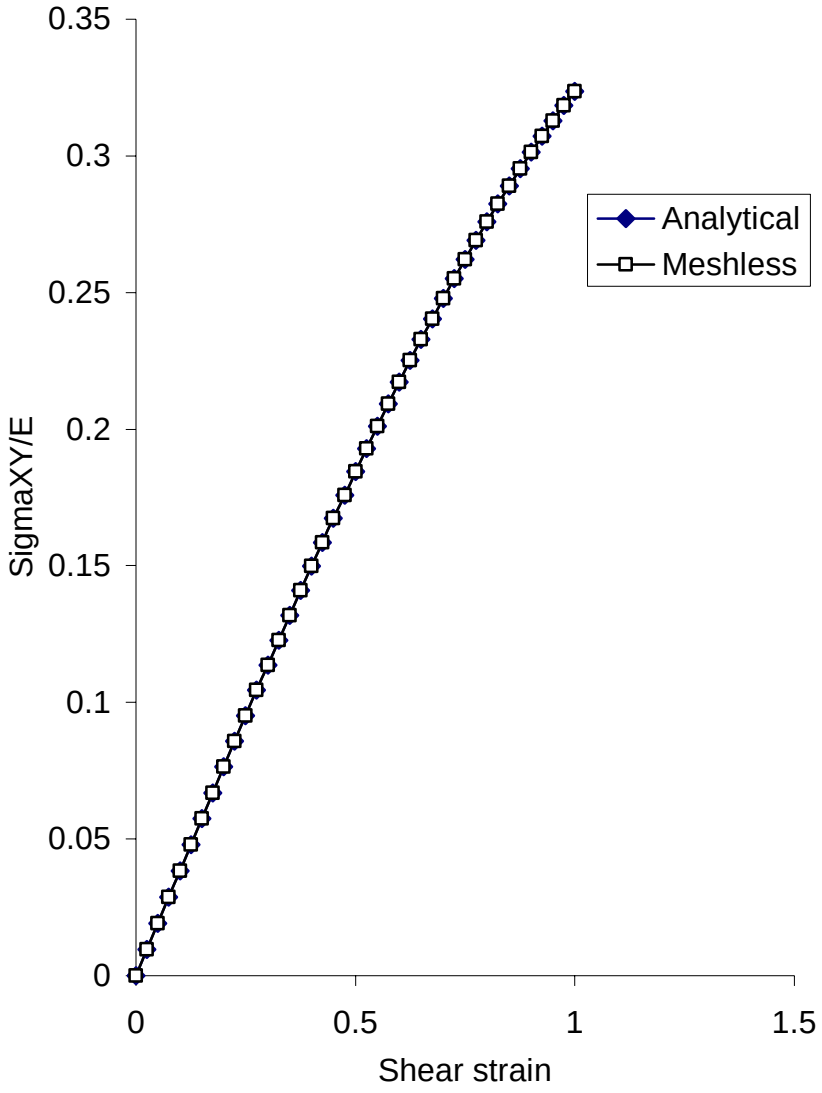


Figure 4.16 FEM results versus meshless results for pure shear simulation for elastoplastic case

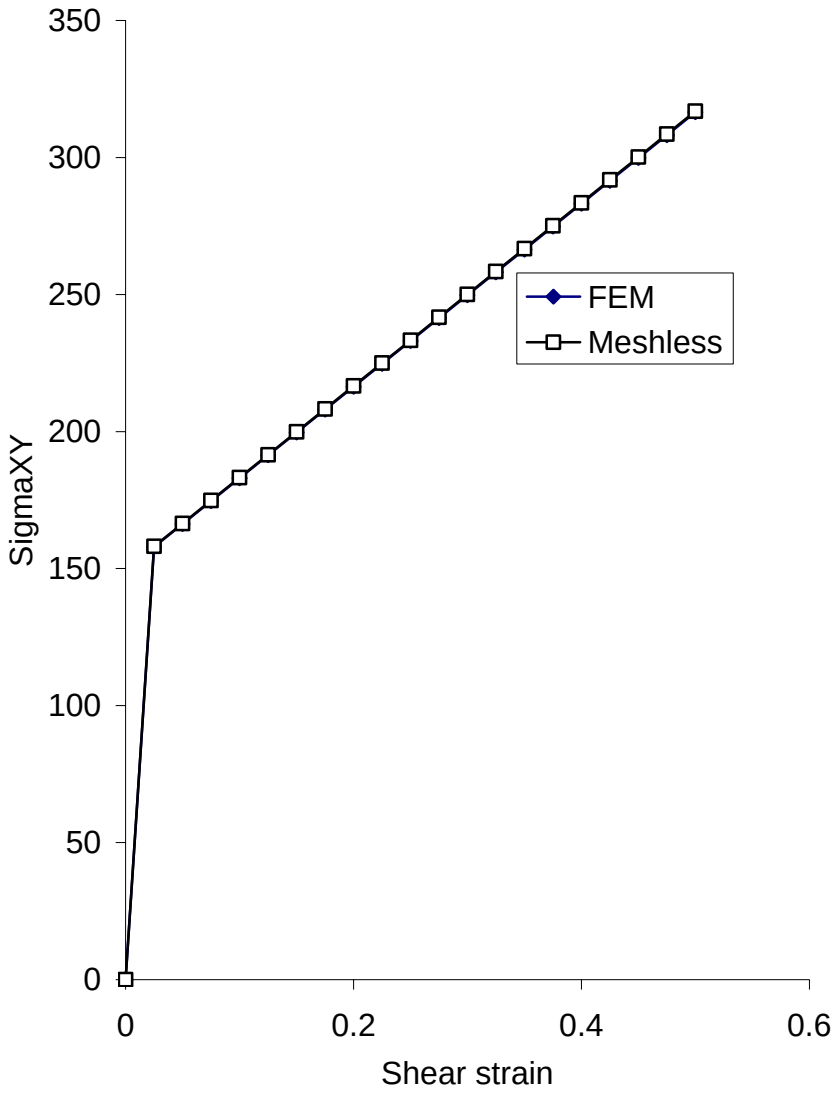
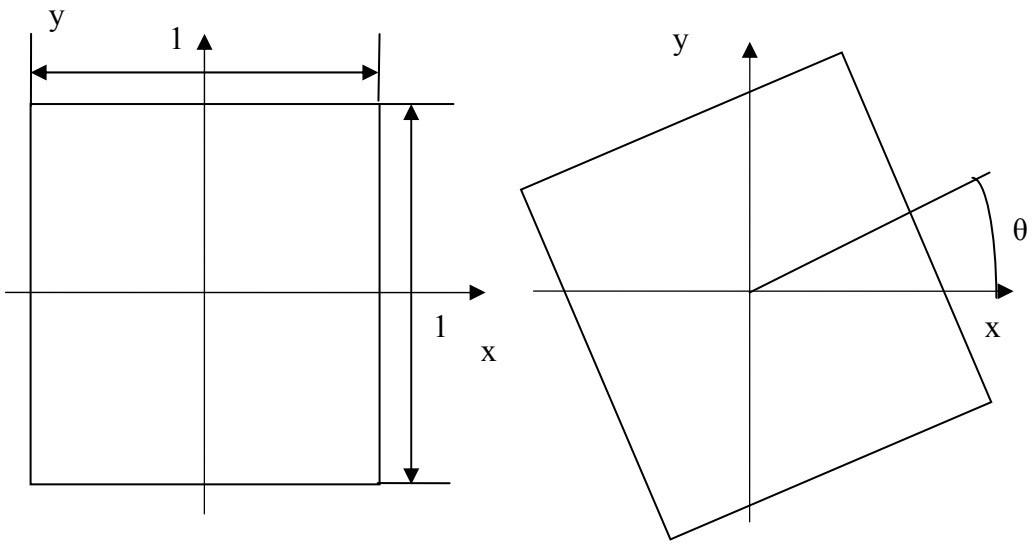


Figure 4.17 Rotation of a prestressed square with no deformation. (a) Original configuration. (b) Current configuration.



(a) Original configuration (b) Current configuration

Figure 4.18 Analytical versus meshless results (σ_{11}) for rotation test for different rotation theta

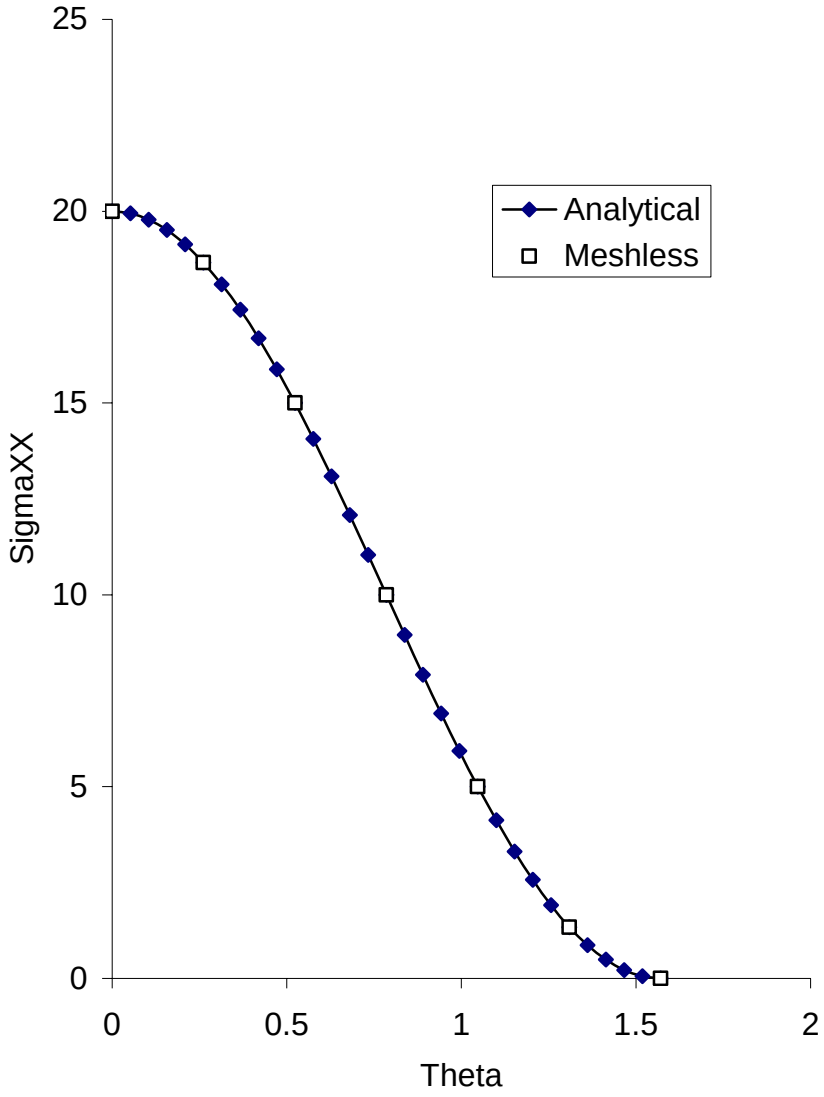


Figure 4.19 Analytical versus meshless results (σ_{12}) of the rotation test for different rotatation theta

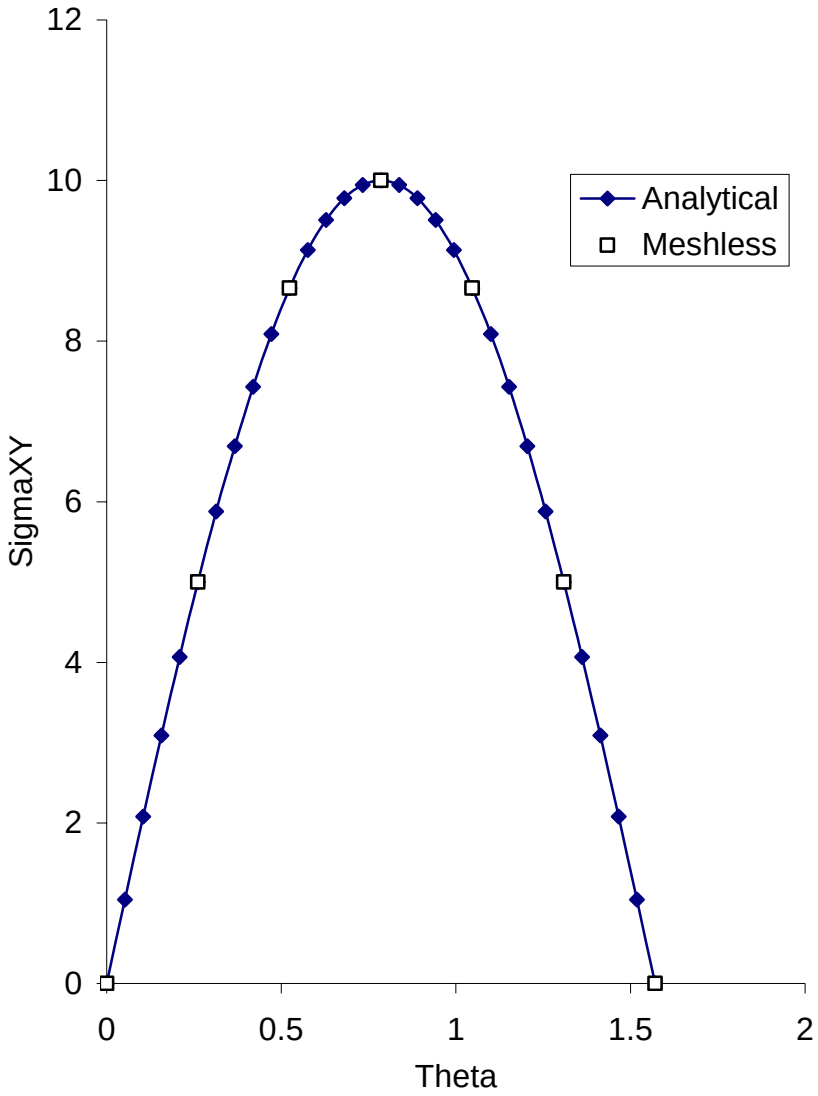


Figure 4.20 Analytical versus meshless results (σ_{22}) of the rotation test for different rotation theta

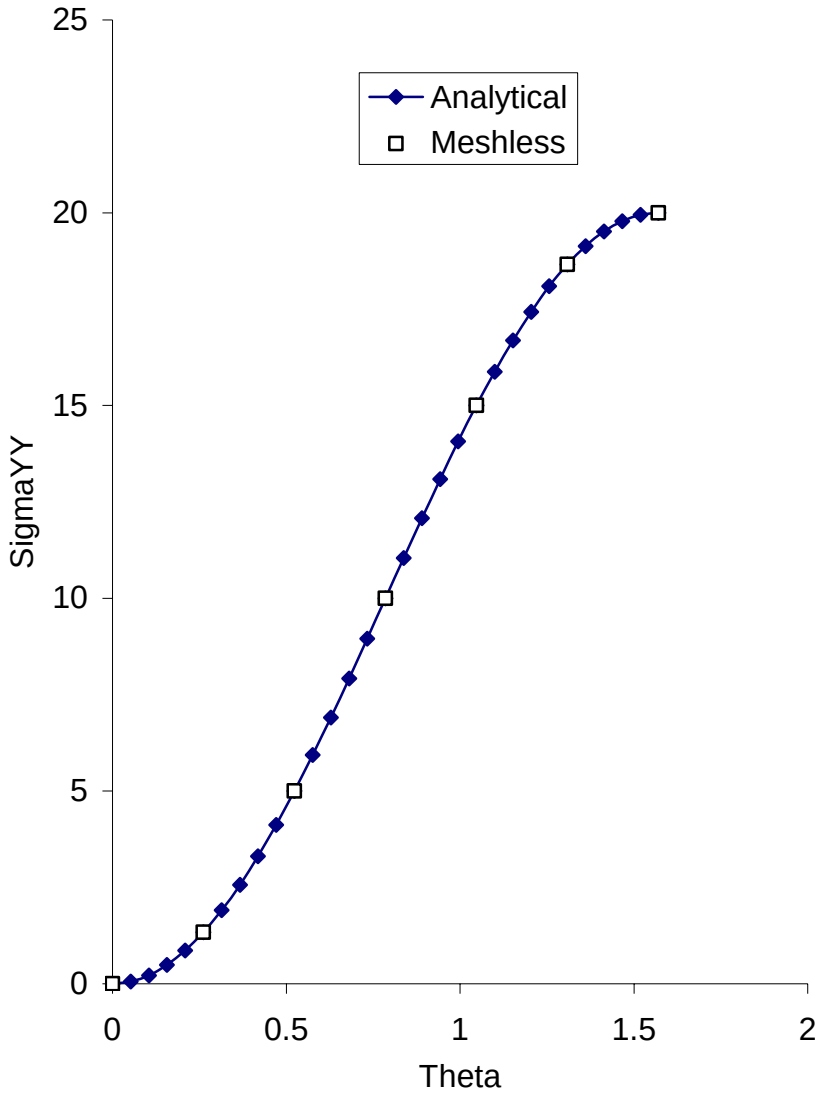


Table 4.1 Special numerical integration for functions containing logarithmic singularity (8 integration points)

$$\int_0^1 f(r)\ln\left(\frac{1}{r}\right)dr = \sum_{i=1}^{N} w_i f(r_i)$$

Abscissas (r_i)	Weights (w_i)
0.0133 2024 4160 8925	0.1644 1660 4728 0030
0.0797 5042 9013 8949	0.2375 2561 0023 3060
0.1978 7102 9326 1880	0.2268 4198 4431 9190
0.3541 5399 4351 9090	0.1757 5407 9006 0700
0.5294 5857 5234 9170	0.1129 2403 0246 7590
0.7018 1452 9939 1000	0.0578 7221 0717 7821
0.8493 7932 0441 1070	0.0209 7907 3742 1330
0.9533 2645 0056 3600	0.0036 8640 7104 0276

Table 4.2 The comparison between hand calculation solution and meshless result for Ux=0.1 (1 load increment)

	σ_{11}	σ_{33}	b
Hand solution	23423	7026	-0.046061
Meshless	23423	7026	-0.046061

Table 4.3 The comparison between hand calculation solution and meshless result for Ux=0.1 (2 load increments)

	σ_{11}	σ_{33}	b
Hand solution	22297	6696	-0.042859
Meshless	22297	6696	-0.042859

CHAPTER 5 - A Truly Meshless Pre- and Post-Processor for Meshless Analysis Methods

Published in:

Advances in Engineering Software 2007; 38: 9-30

Author's Names:

Jianfeng Ma, Prakash Krishnaswami, and X.J. Xin

Author's Affiliations

Department of Mechanical and Nuclear Engineering, Kansas State University, 302 Rathbone Hall, Manhattan, KS 66506-5205

5.1 Abstract

In recent years, meshless methods have been developed to eliminate the known drawbacks in finite element methods. Generating the input file for a meshless method and interpreting the output obtained can be difficult without graphical pre-processing and post-processing support. Unfortunately, most existing pre- and post-processing techniques are based on using an underlying finite element mesh or finite difference grid. Since meshless methods have neither, new approaches are required for providing this support for meshless methods. In this paper, a pre-processor and a post-processor are presented for the meshless method using node-based and pixel-based approaches as opposed to an element-based approach. Pre-processing supports for automated generation of nodes, support domains, and sub-domains along with local refining are also included. An extensive example is presented to demonstrate the effectiveness of the given pre-processor and post-processor.

KEY WORDS: meshless method, meshless post-processor, pre-processor, post-processor, linear elasticity

5.2 Introduction

The development of approximate methods for the numerical solutions of partial differential equations has attracted the attention of engineers, physicists and mathematicians for a long time. In recent years, meshless methods have been developed as an alternative numerical approach to eliminate known drawbacks in the well-known finite element method (FEM). Meshless methods do not require a mesh to discretize the problem domain, and the approximate solution is constructed entirely based on a set of scattered nodes. Meshless methods may also alleviate some other problems associated with FEM, such as element distortion, locking, and remeshing during large deformations. Moreover, nodes can be easily added or removed. As the deformation progresses, more accurate solutions can be obtained using meshless methods.

Several versions of meshless methods have been developed, which may be broadly divided into two categories: boundary type methods such as the boundary node method (BNM) [1][2] and boundary point interpolation method (BPIM) [3]; and domain type methods such as diffuse element method (DEM) [4], element free Galerkin (EFG) method [5], reproducing kernel particle method (RKPM) [6][7][8], point interpolation method (NM) [9], and point assembly method (PAM) [10]. Most methods are "meshless" only in terms of the interpolation or approximation of the field or boundary variables, as compared to the usual boundary element method (BEM) or finite element method (FEM), but still have to use a background mesh to integrate a weak form over the problem domain or the boundary. The need for a background mesh for integration makes these methods not truly meshless.

Some truly meshless methods have been developed including the meshless local Petrov-Galerkin (MLPG) method [11][12][13][14], the local boundary integral equation (LBIE) method [15][16][17][18][19] and the local point interpolation method (LPIM) [20]. These methods do not require a background mesh either for purposes of interpolation of the solution variables, or for the integration of the energy. To use a meshless method, the input file describing the problem of interest must be constructed; further the user may also need support for analyzing and visualizing the results obtained by the meshless method. There are commercial software packages that have these two functionalities, such as ANSYS [21], MSC/PATRAN [22] and CONPLOT [23], but these are element-based software packages designed for use with FEM solvers; for meshless methods, because there is no element at all, we cannot use these software

packages. Liu [24] developed a post-processor for one meshless method, but this is element-based and cannot be used by other meshless methods.

In order to make use of the meshless method [19] more easily and for post-processing the output file of the meshless method more conveniently and efficiently, we have developed a generic meshless pre-processor and post-processor for use with meshless solvers. This pre-processor and post-processor can be used generally with any meshless method.

5.3 A meshless method based on regularized boundary integral equation

5.3.1 Regularized boundary integral equation

A meshless integral method based on the regularized local boundary integral equation approach has been developed by the authors [19]. The method is an improved version of the LBIE method proposed previously by Atluri and coworkers [15, 16]. The most critical improvement is the use of the subtraction technique to remove the strong singularity that results in a regularized governing integral equation. A special numerical integration is employed for the calculation of integrals with weak singularity which further improves accuracy. The collocation method is employed to enforce the essential boundary conditions exactly. Details of the meshless integral method have been presented in [19]; only a concise description of the method is given here.

The problem domain of interest is an elastic body Ω with boundary Γ. The local boundary integral equation approach seeks to establish the governing equation over a local sub-domain $\Omega^{<a>}$. For simplicity, $\Omega^{<a>}$ is chosen as a sphere (or part of a sphere) with radius $h_S^{\langle a\rangle}$ centered on a node $\mathbf{y}^{\langle a\rangle}$. If $\mathbf{y}^{\langle a\rangle}$ is an interior node, $h_S^{\langle a\rangle}$ is selected such that $\Omega^{\langle a\rangle}$ stays fully inside the problem domain Ω. If $\mathbf{y}^{\langle a\rangle}$ is a boundary node, $\Omega^{\langle a\rangle}$ is the intersection of Ω and a sphere $S^{<a>}$ of radius $h_S^{\langle a\rangle}$ centered on $\mathbf{y}^{\langle a\rangle}$, and the boundary of $\Omega^{\langle a\rangle}$ is the union of the boundary of $S^{<a>}$ inside Ω, denoted as $C^{<a>}$, and the boundary of Ω inside $S^{<a>}$, denoted as $\Gamma_u^{<a>}$ and $\Gamma_t^{<a>}$ where $\Gamma_u^{<a>}$ and $\Gamma_t^{<a>}$ are respectively the prescribed displacement and prescribed traction boundary.

Applying the weak form of equilibrium over the local sub-domain $\Omega^{<a>}$ and employing a singularity removal procedure, the regularized local boundary integral equation is obtained as follows [19]:

$$\begin{aligned}\alpha_{ij}^{\langle a\rangle}u_j^{\langle a\rangle} = & \int_{\Gamma_u^{\langle a\rangle}+\Gamma_t^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)t_j(\mathbf{x})\,d\Gamma - \int_{C^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)u_j(\mathbf{x})\,d\Gamma \\ & - \int_{\Gamma_u^{\langle a\rangle}+\Gamma_t^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left(u_j(\mathbf{x})-u_j^{\langle a\rangle}\right)d\Gamma + \int_{\Omega^{\langle a\rangle}} b_j(\mathbf{x})\,\tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)d\Omega\end{aligned} \tag{1}$$

where $u_j(\mathbf{x})$ and $u_j^{<a>}$ are the j-th displacement components of point $\mathbf{x}$ and of node $\mathbf{y}^{<a>}$ respectively, $t_j(\mathbf{x})$ is the j-th traction component associated with $u_j(\mathbf{x})$, and $b_j(\mathbf{x})$ is the j-th body force component. $\tilde{u}_{ij}^*$ and $\tilde{t}_{ij}^*$ are special test functions. Expressions for $\tilde{u}_{ij}^*$, $\tilde{t}_{ij}^*$ and $\alpha_{ij}^{<a>}$ have been given in [19]. In Equation (1), the subtraction method [19] has been used in the third term of the right hand side which removes the strong singularity and significantly enhances the accuracy and efficiency of the method.

5.3.2 Meshless Implementation

Equation (1) holds for any source node $\mathbf{y}^{\langle a\rangle}$, either inside the domain or on the boundary. The general solution strategy is to select a set of source nodes and apply Equation (1) to each of these source nodes to obtain an equal number of governing equations. The moving least squares approximation (MLSA) (see for example [5]) is employed to approximate the displacement field in terms of nodal values at the source nodes. Numerical quadrature is then applied to evaluate the integrals, which converts the integral equations into a system of simultaneous algebraic equations. If all source nodes are selected on the boundary, a (regularized) boundary element method will result. In the meshless integral method, both boundary and interior source nodes are used, leading to a full domain method.

A ***support domain*** for node $\mathbf{y}^{\langle a\rangle}$ is defined as a sphere (3D) or disk (2D) centered on $\mathbf{y}^{\langle a\rangle}$ with a radius $\ell_w^{<a>}$. A weight function $w^{\langle a\rangle}$ is a continuous function that is positive in the support domain and zero outside, i.e.

$$\begin{cases} w^{\langle a\rangle}(\mathbf{x}) \geq 0 & \text{if } \left|\mathbf{x}-\mathbf{y}^{\langle a\rangle}\right| \leq \ell_w^{\langle a\rangle} \\ w^{\langle a\rangle}(\mathbf{x}) = 0 & \text{if } \left|\mathbf{x}-\mathbf{y}^{\langle a\rangle}\right| > \ell_w^{\langle a\rangle} \end{cases} \tag{2}$$

As introduced previously, the ***sub-domain*** $\Omega^{\langle a\rangle}$ for node $\mathbf{y}^{\langle a\rangle}$, located entirely inside Ω, is a sphere or part of a sphere centered on $\mathbf{y}^{\langle a\rangle}$ with a radius $h_S^{\langle a\rangle}$. Figure 5.1 illustrates the meaning of local sub-domain and support domain. The concepts of sub-domain and support domain for each node are used in most meshless methods. Thus, a pre-processor must provide support for rapid definition of the sub-domain radius and support domain radius for each node, in addition to automated node generation.

Two other frequently used concepts are the domain of definition and the domain of influence. The domain of definition of a point $\mathbf{x}$ is the set of all nodes whose weight functions are non-zero at $\mathbf{x}$, while the domain of influence of a node $\mathbf{y}^{<a>}$ is the set of all nodes whose weight functions are non-zero in some part or all of the sub-domain of node $\mathbf{y}^{<a>}$. The domain of definition and the domain of influence are convenient terms in the description of MLSA and local boundary integrals, and are illustrated in Figure 5.2.

With MLSA, the displacement at an arbitrary point **x** is approximated by shape functions and nodal displacements as follows

$$\mathbf{u}_h(\mathbf{x}) = \mathbf{\Phi}^T(\mathbf{x})\hat{\mathbf{u}} = \sum_{a=1}^{Nx} \varphi^{<a>}(\mathbf{x})\hat{\mathbf{u}}^{<a>} \tag{3}$$

where N_x is the total number of nodes in the domain of definition of $\mathbf{x}$, and $\phi(\mathbf{x})$ is the shape function. Detailed expression for $\phi(\mathbf{x})$ has been given in [19].

With the above discretization and the conditions that $u_j = \bar{u}_j$ on $\Gamma_u^{<a>}$ and $t_j = \bar{t}_j$ on $\Gamma_t^{<a>}$, Equation (1) becomes (there is a summation on b and j but not on a and i):

$$\sum_{b=1}^{N_y}(H_{ij}^{<a,b>}\hat{u}_j^{<b>}) - \sum_{b=1}^{N_y}(L_{ij}^{<a,b>}\hat{u}_j^{<b>}) + \alpha_{ij}^{<a>}u_j^{<a>} = G_i^{<a>} \tag{4}$$

where $H_{ij}^{\langle a,b\rangle}$, $L_{ij}^{<a,b>}$ and $G_i^{\langle a\rangle}$ are:

$$\begin{aligned} H_{ij}^{\langle a,b\rangle} = & -\int_{\Gamma_u^{\langle a\rangle}} \tilde{u}_{ik}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left[\mathbf{NCB}^{\langle b\rangle}(\mathbf{x})\right]_{kj} d\Gamma + \int_{C^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\phi^{\langle b\rangle}(\mathbf{x})d\Gamma \\ & + \int_{\Gamma_t^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left(\phi^{\langle b\rangle}(\mathbf{x})\right)d\Gamma \end{aligned} \tag{5}$$

$$L_{ij}^{<a,b>} = \int_{\Gamma_t^{<a>}} \tilde{t}_{ij}^*(\mathbf{x},\mathbf{y}^{<a>})(\phi^{<b>}(\mathbf{y}^{<a>}))d\Gamma \tag{6}$$

$$G_i^{\langle a\rangle} = \int_{\Omega^{\langle a\rangle}} b_j(\mathbf{x})\tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)d\Omega + \int_{\Gamma_t^{\langle a\rangle}} \tilde{u}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\bar{t}_j(\mathbf{x})d\Gamma - \int_{\Gamma_u^{\langle a\rangle}} \tilde{t}_{ij}^*\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)\left(\bar{u}_j(\mathbf{x}) - \bar{u}_j\left(\mathbf{y}^{\langle a\rangle}\right)\right)d\Gamma \tag{7}$$

and the upper limit of summation, $N_{\mathbf{y}}$, is the total number of nodes in the domain of influence of node $\mathbf{y}^{<a>}$. In the present study, the body force is neglected. The integrands in the first integral of (5) and the second integral in (7) contain the weakly singular ln(r) term. A special integration scheme for the logarithmic singularity [26] is employed in the meshless integral method that was shown to achieve excellent numerical accuracy [19].

5.3.3 Imposition of Essential and Natural Boundary Conditions

Appropriate boundary conditions need to be imposed in order to solve the simultaneous Equations (4). In meshless methods, imposing the essential (Dirichlet) boundary conditions is not as trivial as in the finite element method. Because MLSA is non-interpolative, the essential boundary condition does not take the form of prescribed value for the fictitious nodal displacement ($\hat{u}_i^{<a>} = \bar{u}_i^{<a>}$), but rather a constraint equation involving a linear combination of the fictitious nodal displacements in a neighborhood of the boundary node. A number of techniques for the imposition of essential boundary conditions have been developed, including: (1) collocation methods [27]; (2) Lagrange multiplier method [5]; (3) penalty method [28]; (4) Nitsche's method [29]; (5) coupled meshless-finite element method [30]; (6) method based on d'Alembert principle [31]; (7) use of window or correction functions that vanish on the boundary [32]; (8) discrete form of essential boundary conditions [33]; and (9) displacement constraint equation method [34]. The advantages and disadvantages of each of these methods have been discussed briefly in [19].

Collocation methods for enforcing essential boundary conditions are defined as those methods in which conditions are enforced exactly at a discrete set of boundary nodes [35]. A number of collocation methods have been developed. The direct collocation method [27] used the collocation condition

$$\hat{u}_i^{<a>} = \bar{u}_i^{<a>} \tag{8}$$

to replace the row of the discretized weak form equation corresponding to the degree of freedom with prescribed displacement $\bar{u}_i^{<a>}$. The direct collocation is actually inconsistent with the assumption of MLSA since the fictitious nodal displacement $\hat{u}_i^{<a>}$ is generally not equal to the approximated displacement value. A modified collocation method uses

$$u_i^{\langle a\rangle} = \sum_{b=1}^{n} \phi^{\langle b\rangle}\left(\mathbf{y}^{\langle a\rangle}\right)\hat{u}_i^{\langle b\rangle} = \bar{u}_i^{\langle a\rangle} \tag{9}$$

as the collocation condition which was shown to yield more accurate results [36].

Wagner and Liu [35] pointed out that when the shape functions do not satisfy the Kronecker delta property, the rows of the matrix equation corresponding to the essential boundary nodes contribute to the solution of the displacement field, and that when these equations are simply ignored, as is done in the traditional (including direct and modified) collocation method, the weak form is not satisfied consistently. Such inconsistency leads to poor accuracy and low convergence rate. Wu and Plesha [37] proposed a boundary flux collocation method to enforce the boundary conditions exactly which maintains the consistency of the weak form and avoids partitioning and rearrangement of the nodal DOFs into a constrained and a free group as needed in the corrected collocation method.

Generally, there are two types of discretization in meshless methods: (1) Galerkin based discretization over the global domain; and (2) discretization over multiple local domains. For global domain-based discretization, the n system equations are obtained by applying the weak form over the global domain once, and therefore all equations must hold simultaneously in order to maintain consistency in the weak form. Replacing a row in the matrix equation by (9), which contains a linear combination of DOFs rather than dictating the value of the single constrained DOF, sacrifices the consistency of the weak form and compromises the accuracy of the solution. The inconsistency discussed in [35][37] therefore pertains to the global domain-based discretization methods only. For local domain-based discretization, each equation is obtained by applying the weak form over a particular local domain, and the weak form needs to be applied n times for a problem with n DOFs. Consequently, the modified collocation method with (11) can be used easily and directly to impose essential boundary conditions, because each of the system equations is independent of the rest, and replacing the equation corresponding to a constrained DOF by (9) will not cause any inconsistency in the weak form.

The current meshless integral method utilizes the local domain-based discretization, and the modified collocation method can be directly used to impose essential boundary conditions. Since the system equations are obtained by applying the integral Equation (1) to each source node over a local sub-domain, for a DOF with essential boundary condition, we simply replace the governing equation corresponding to the DOF with essential boundary condition (9). Numerical verifications in [19] show that the modified collocation method for imposing essential boundary conditions works well with the meshless integral method.

For the natural boundary condition $t_i = \bar{t}_i$, no special treatment is needed. The prescribed traction is directly used in the second integral in Equation (7).

After the essential and natural boundary conditions are imposed, the governing equations are

$$\sum_{b=1}^{N_y} (K_{ij}^{<a,b>} \hat{u}_j^{<b>}) = F_i^{<a>} \tag{10}$$

where

$$K_{ij}^{<a,b>} = \begin{cases} H_{ij}^{<a,b>} - L_{ij}^{<a,b>} + \alpha_{ij}^{<a>} \phi^{<b>}(\mathbf{y}^{<a>}) & \text{when } u_i^{<a>} \text{ is unconstrained} \\ \phi^{<b>}(\mathbf{y}^{<a>}) \delta_{ij} & \text{when } u_i^{<a>} = \bar{u}_i^{<a>} \end{cases} \tag{11}$$

$$F_i^{<a>} = \begin{cases} G_i^{<a>} & \text{when } u_i^{<a>} \text{ is unconstrained} \\ \bar{u}_i^{<a>} & \text{when } u_i^{<a>} = \bar{u}_i^{<a>} \end{cases} \tag{12}$$

and the upper limit of summation, N_y, is the total number of nodes in the domain of influence of node $\mathbf{y}^{<a>}$.

After we get a fictitious displacement ($\hat{\mathbf{u}}$) for each node, by using Equations (13) through (16) we can get the actual displacements and other variables (strains ($\boldsymbol{\varepsilon}$) and stresses ($\boldsymbol{\sigma}$)) for every point between these nodes. For each point, only the nodes whose weight functions are not zero at this point can contribute in the computation of these variables.

$$u_j(\mathbf{x}) = \sum_{b=1}^{Nx} \varphi^{\langle b \rangle}(\mathbf{x}) \hat{u}_j^{\langle b \rangle} \tag{13}$$

$$u_{j,k}(\mathbf{x}) = \sum_{b=1}^{Nx} \varphi_{,k}{}^{\langle b \rangle}(\mathbf{x}) \hat{u}_j^{\langle b \rangle} \tag{14}$$

True strains and stress are calculated by:

$$\begin{Bmatrix} \varepsilon_{11}(\mathbf{x}) \\ \varepsilon_{22}(\mathbf{x}) \\ 2\varepsilon_{12}(\mathbf{x}) \end{Bmatrix} = \sum_{b=1}^{N_x} [\mathrm{B}^{<b>}(\mathbf{x})] \begin{Bmatrix} \hat{u}_1^{<b>} \\ \hat{u}_2^{<b>} \end{Bmatrix} \tag{15}$$

$$\begin{Bmatrix} \sigma_{11}(\mathbf{x}) \\ \sigma_{22}(\mathbf{x}) \\ \sigma_{12}(\mathbf{x}) \end{Bmatrix} = \sum_{b=1}^{N_x} [\mathrm{CB}^{<b>}(\mathbf{x})] \begin{Bmatrix} \hat{u}_1^{<b>} \\ \hat{u}_2^{<b>} \end{Bmatrix} \tag{16}$$

Where N_x is the total number of nodes in the domain of definition of point $\mathbf{x}$. Detailed expressions for C and $\mathbf{B}^{\langle b \rangle}$ have been given in [19]

The post-processor that we have developed assumes that the meshless method outputs the fictitious displacements to a file that the post-processor reads and uses for interpolation. Generally, most meshless methods use this approach of calculating fictitious displacements as a way to calculate the actual values of field variables at points in between nodes. The post-processor presented in this paper also uses this idea, and so it should be compatible with most meshless solvers.

5.4 The Pre-processor

The pre-processor is used to define the data and discretization scheme for meshless analysis. This includes support for the geometric model, as well as definition of material, boundary conditions, and other parameters for the meshless method.

One salient feature of this pre-processor is that the troublesome and time-consuming node generation process is automated. For discretization of a problem domain, the pre-processor can construct nodal information for the entire domain with minimal user input. This significantly reduces the manpower required for node generation. The pre-processor also provides a high degree of automation in the generation of the sub-domain and support domain. The user can enter all required information using graphical user interfaces (GUI), without the need to create any data file. A typical GUI element used in the pre-processor is shown in Figure 5.3a. This dialog is used for defining the material properties and other modeling parameters. Similarly, essential boundary conditions and natural boundary conditions on a selected boundary node can be specified through the dialogs shown in Figure 5.3b and Figure 5.3c. In similar fashion, all required data can be supplied through the GUI, without having to create any data files manually.

Based on this information, the pre-processor automatically generates the nodes, as well as the sub-domain and support domain associated with each node as described below.

5.4.1 Node Generation

The node generation that is supported is based on subdivision of the problem domain into quadrilateral. In this approach, the user first divides the problem domain into quadrilaterals using the GUI. For each of these quadrilaterals, we use the quadrilateral-based method described in Algorithms 5.4.1 and 5.4.2 to generate the nodes. The user may specify node density or use the pre-processor's default values. These algorithms are presented below.

Algorithm 5.4.1: Generation of nodes within a given quadrilateral domain

Given: Edges E(i), i=0,1,…,3 of the quadrilateral; type of each edge Type(j), j = 0,…,3 (the value of each Type(j) can only be LINE or ARC); Start pt. $(x_s(i), y_s(i))$, end pt. $(x_e(i), y_e(i))$, center pt. $(x_c(i), y_c(i))$ for each edge i, i = 0,…3 (center pt. is ignored for an edge which is a LINE); N_1, the no. of divisions for 1st and 3rd edges, and R_1 , the corresponding division ratio; N_2, the no. of divisions for 2nd and 4th edges, and R_2, the corresponding division ratio.

Output: Locations Loc(N+k), k = 0, ….., $(N_1+1)*(N_2+1)-1$ of the nodes generated; N is the initial size of Loc.

/

Note: Each Loc(k) is a location, i.e. an (x, y) coordinate pair. The following functions are assumed to be available:

***crossproduct((x_1,y_1), (x_2,y_2))**: Returns the cross product of two vectors.*

***Intersection(L_1, L_2)**: Returns the intersection of the two straight lines L_1 and L_2 .*

**/*

Step 1: for (i = 0; i < 4; i++)

if ((i==0)||(i==2)) { $q = e^{\frac{\ln R_1}{N_1 - 1}}$; $N_0 = N_1$; }

else {$q = e^{\frac{\ln R_2}{N_2 - 1}}$; $N_0 = N_2$; }

endif;

if (Type(i)==LINE); *//Starting pt. $(x_s(i), y_s(i))$; ending pt. $(x_e(i), y_e(i))$*

if(q==1) //uniform distribution of nodes

UnitX = $(x_e(i)-x_s(i))/N_0$; UnitY=$(y_e(i)-y_s(i))/N_0$;

Pts(i,N_0) = $(x_e(i), y_e(i))$;

Pts(i,j)=$(x_s(i)+j*\text{UnitX}, y_s(i)+\text{UnitY})$, j = 0, …, N_0-1;

//Pts are pts. on edge i.

else *//non-uniform distribution of nodes*

UnitX=$(x_e(i)-x_s(i))*(1-q)/(1-q^{N_0})$;UnitY=$(y_e(i)-y_s(i))*(1-q)/(1-q^{N_0})$;

Pts(i,0)= $(x_s(i), y_s(i))$;

Pts(i,j)=$(x_s(i)+\text{UnitX}*(1-q^j)/(1-q), y_s(i)+\text{UnitY}*(1-q^j)/(1-q))$,

j = 1,…, N_0-1;

Pts(i,N_0)= $(x_e(i), y_e(i))$;

end if *// end of if-else block on q*

end if *// end of if block for (Type(i)=LINE)*

if(Type(i)==ARC) *// Starting pt. ($x_s(i)$, $y_s(i)$); ending pt. ($x_e(i)$, $y_e(i)$);*
// center ($x_c(i)$,$y_c(i)$);

$R=\sqrt{(x_s(i)-x_c(i))^2+(y_s(i)-y_c(i))^2}$;

$x_1 = x_s(i)-x_c(i)$; $y_1 = y_s(i)-y_c(i)$; *//(x_1,y_1)vector of start pt. from center*

$x_2 = x_e(i)-x_c(i)$; $y_2 = y_e(i)-y_c(i)$; *//(x_2,y_2)vector of end pt. from center*

refX = $x_c(i)-x_1$; refY=y_c-y_1;

$\alpha = \cos^{-1}(\frac{x_1 * x_2 + y_1 * y_2}{\sqrt{x_1^2+y_1^2} * \sqrt{x_2^2+y_2^2}})$;*//dotproduct((*x_1,y_1*), (*x_2,y_2*))*

$\alpha\_ref = \cos^{-1}(\frac{x_s(i)}{\sqrt{x_s(i)^2+y_s(i)^2}})$; *//dotproduct((*x_1,y_1*), (1,0))*

if (crossproduct((x_2,y_2), (-x_1,-y_1))<0) $\alpha = 2*\pi-\alpha$;

if (crossproduct((x_1,y_1),(-1,0))<0) $\alpha\_ref = 2*\pi-\alpha\_ref$;

if (q==1) *// uniform distribution of nodes*

unitAngle=$\frac{\alpha}{N_0}$; Pts(i,0) = ($x_s(i)$, $y_s(i)$);

Pts(i,j) = ($x_c(i)$+R*cos(α_ref +unitAngle*i),
$y_c(i)$+R*sin(α_ref +unitAngle*i)),
j = 1, …, N_0-1;

Pts(i,N_0) = ($x_e(i)$, $y_e(i)$);

else *//non-uniform distribution of nodes*

unitAngle = α *(1-q)/(1-q^{N_0});

Pts(i,0) = ($x_s(i)$, $y_s(i)$); Pts(i,N_0) = ($x_e(i)$, $y_e(i)$);

Pts(i, j) = ($x_c(i)$+R*cos(α_ref +unitAngle*(1-q^j)/(1-q)),
$y_c(i)$+R*sin(α_ref +unitAngle*(1-q^j)/(1-q)),
j = 1, …, N_0-1;

end if *// end of if-else block for q*

end if *// end of if block for (Type (i) == ARC)*

end for *// end of for loop on i*

Step 2: $L_1(i)$ = Line(Pts(0, i), Pts(2, i)), i = 0,1,…., N_1; *//Line from node i of E_1 to*
// node i of E_3.

$L_2(j)$ = Line(Pts(1, j), Pts(3, j)), j = 0,1,…., N_2; *// similarly for E_2 and E_4.*

Step 3: k=0;

for (i=0; i<N_1+1; i++)

for (j=0; j<N_2+1; j++)

Loc(N+k)=Intersection($L_1(i)$, $L_2(j)$);

k=k+1;

end for

end for

Step 4: N=N+(N_1+1)*(N_2+1); *//update the size of the locations*

// End of Algorithm 5.4.1

The set of nodes generated by applying Algorithm 5.4.1 will generally contain duplicates. This is because we generate the nodes quadrilateral by quadrilateral; thus, nodes that lie on an edge that is common to two quadrilaterals will be generated twice. To resolve this difficulty, we must scan through all the nodes generated by Algorithm 5.4.1 and eliminate duplicates. This will yield a reduced set of nodes that can be directly used as a valid discretization of the domain. The method that was developed for identifying and eliminating duplicates is detailed in Algorithm 5.4.2.

Algorithm 5.4.2: To generate a reduced set of nodes from the set of nodes generated by Algorithm 5.4.1 and set the type of each node in the reduced set

Given: Ω={Loc(i), i=0,1,…N-1}, the set of nodes generated by Algorithm 3.3 for all quadrilateral of the problem domain; the global boundary segments {S(k), k=0,1,…M-1} where M is the number of segments; the start point ($x_s(k)$, $y_s(k)$) and the end point ($x_e(k)$, $y_e(k)$) of the k-th segment.

To find: the reduced set of nodes, Ω_1={loc(j), j=0,1,... N_1-1},eliminating duplicates from Ω.

/*

*The initial size $N_1$is 0, and the type of each node **type(j)**, size is N_1 (if loc(j) is boundary node, type(j)=1 otherwise type(j)=0). If loc(j) is boundary node, also specify the two outward normals of this node: **leftNormal(j)**, **rightNormal(j)**.*

*See Algorithm 3.3 for explanation of **leftNormal** and **rightNormal**.*

***getNormalSeg(k,(x,y))**: Return the outward normal of segment k at point (x, y).*

**/*

Step 1: $N_1 = 0$

```
            for (i=0; i<N; i++)
        coincide = false; //flag for judging if this loc is already in Ω1
        for (k=0; k<N1; k++)
            if (Loc(i) == loc(k))  {coincide = true;     break;}
            end if
                end for // end of for loop on k
        if (coincide==false)   {loc( N1 ) = Loc(i);  N1 =N1+1;}
        end if //end of if block on coincide
      end for // end of for loop on i
```

Step 2:

```
    for (i=0; i<N1; i++)
        flag = false; // judge if loc(i) is on boundary
        for (j=0; j<M; j++)
            if (loc(i) is on S(j));
                flag=true;
                if (loc(i) == (xs(j),ys(j)))
                    leftNormal(i) = getNormalSeg(j,(xs(j),ys(j)));
                else if (loc(i) == (xe(j),ye(j))) //if loc(i) is ending pt. of S(j)
                    rightNormal(i) = getNormalSeg(j, (xe(j),ye(j));
                else // smooth boundary location
                    leftNormal(i) = getNormalSeg(j, loc(i));
```

```
                rightNormal(i) = getNormalSeg(j, loc(i));
            end if // end of if-else block for the relationship between node and
                //boundary.

        end if
    end for // end of for loop for j
    if (flag==true) type(i) = 1;
    else type(i) = 0;
    end if // end of if-else block for flag
  end for // end of for loop for i
// End of Algorithm 5.4.2
```

5.4.2 Node Refinement

After we get the reduced set of nodes for the given problem domain, the user may want to refine the nodes in some local part of the problem domain. This will generally result in an irregular mesh that is better suited for the problem at hand. This pre-processor provides the functionality for node refinement in order to generate irregularly distributed nodes. The pre-processor supports refinement of interior nodes as well as boundary nodes.

Internal refinement enables the user to refine the nodes in a given part of the problem domain. Figures 5.4a to 5.4d illustrate how the pre-processor does this. Figure 5.4a shows the original node set and the problem domain. The user first specifies the rectangle within which the nodes are to be refined. This rectangle may extend beyond the problem domain. As shown in Figure 5.4b, the pre-processor removes all the existing nodes located within the rectangle. The pre-processor then generates a new set of nodes in the given rectangle using Algorithm 5.4.1, based on the new nodal density specified by the user (Figure 5.4c). Figure 5.4d illustrates the final nodal distribution after refinement is completed. It should be noted that the use may specify varying node spacing on along the x and y directions.

The user can also refine the node distribution on any given boundary edge. Once again, the user selects a segment of the boundary edge for refinement and specifies the new node distribution. The existing nodes on this section of the boundary are then cleared and the new set of nodes is created. Figures 5.5a to 5.5c show how the pre-processor does boundary refinement. Figure 5.6a and 5.6b show the dialogs through which the user may enter the required parameters for node refinement.

5.4.3 Sub-domain and Support Domain Generation

After we get the set of nodes, we need to determine two geometric parameters for each node: the sub-domain radius $h_s^{<a>}$ and the support domain radius (or the size of weight function) $\ell_w^{<a>}$. In order to determine the sub-domain radius at each node, we use the following criteria: for each interior node, the sub-domain radius is the minimum distance between the node of interest and all other nodes and all boundary segments; for each corner node (a kind of boundary node), the sub-domain radius is the minimum distance between this node and all boundary segments except the two segments passing through this node; for each smooth boundary node, the sub-domain radius is the minimum distance between this node and all boundary segments along with all ending points of all segments. According to these criteria, the sub-domain of an interior node is always a full sphere (or disk, in 2D) that is centered at this node is fully inside the problem. The sub-domain of a boundary node is always a spherical fan (circular fan in 2D) that is inside the problem domain and is centered at this node. The internal boundary angle subtended by material for each boundary node is detailed in [19]. This approach to generation of sub-domains ensures that we will not have to deal with arbitrarily shaped sub-domains, since the two possibilities listed above are exhaustive. Thus, even for complex problem domains, we will only have to deal with sub-domains that are fully circular or are circular fans. It should be noted that in this approach, the union of all the sub-domains may not cover the total problem domain, and the degree of coverage depends on the node distribution scheme that is used. However, the meshless solver described in Section 2 does not require that the problem domain be covered by the union of the sub-domains, and is capable of generating accurate results with this definition of the sub-domains. A detailed verification of this can be found in [19].

For determining the support domain radius associated with each node, the driving concern is that the support domain should include enough nodes to avoid singularities during the moving least-squares curve fitting process. The minimum number of nodes that each support domain should include depends on the order of the monomial basis; specifically, the support domain should include at least 3 other nodes if we are using a linear basis, at least 6 other nodes if we are using a quadratic basis, and at least 10 other nodes if we are using a cubic basis. To be on the conservative side, we usually define a safety number N_s and impose the requirement that every support domain should contain at least $3+ N_s$ other nodes for a linear basis, $6+ N_s$ other nodes for a quadratic basis, and $10+ N_s$ other nodes for a cubic basis. The safety number N_s is

user selectable and is generally between 5 and 10. It should be noted that it is also undesirable to leave too many nodes within the support domain, since it increases computational effort. Therefore, our approach is to start with a guess for the support domain radius and find the number of nodes it encompasses. If this number is too small, we will increase the radius by some factor and repeat the process; if it includes too many, we reduce the radius by a factor and repeat the process.

The details of the algorithms for automatic generation of sub-domain and support domain radius are given in Algorithm 5.4.3 and Algorithm 5.4.4.

Algorithm 5.4.3: Determine the sub-domain radius for each node.

Given: $\{ y^{<a>}, a=0,1,...N-1\}$ where $y^{<a>}$ represents the location of node a and N is the number of nodes; $\{S(k), k=0,1,...M-1\}$ where $S(k)$ represents the k-th boundary segment and M is the number of segments; the type of each node type(a), a=0,1,...N-1 (if $y^{<a>}$ is a boundary node, type(a)=1 otherwise type(a)=0); the start point $(x_s(k), y_s(k))$ and end point $(x_e(k), y_e(k))$ of the k-th boundary segment, k = 0,1, …, M-1.

/

Every node on the boundary has two outward normals defined, corresponding to the segment to the left of the node and the segment to the right of the node; these are referred to as the ***leftNormal()*** *and* ***rightNormal()****. If the boundary is smooth at a node, then the leftNormal and rightNormal are identical; for corner nodes, leftNormal and rightNormal will be different.*

Note that each node $y^{<a>}$ is a location, i.e., it is an (x, y) coordinate pair. The global boundary is ordered clockwise. Existence of the following functions is assumed:

getLeftNormal(a)*: Return the left outward normal of boundary node $y^{<a>}$.*

getRightNormal(a)*: Return the right outward normal of boundary node $y^{<a>}$.*

dist_to_edge($y^{<a>}$, S(i))*: Returns distance from node $y^{<a>}$ to segment S(i).*

dist (location_1, location_2)*: Return the distance between location_1 to location_2.*

**/*

To find: Sub-domain radii $\{h_s^{<a>}, a = 0,1,...N-1\}$

```
    for (a=0; a<N; a++)
if (a==0) minC=dist (y<0>,y<1>);
        else  minC = dist (y<a>,y<a-1>);
        end if // end of if block for a==0
for (i=0; i<N; i++)
//find the minimum dist. between node a and neighboring nodes
    if ((a!==i) && (dist ( y<a>, y<i> )<minC) ) minC= dist( y<a>, y<i> );
end for // end of the for loop on  i
if (type(a)==0) // interior node
    minD = dist_to_edge(y<a>, S(0));
    for (j=1; j<M; j++)
        if (dist_to_edge(y<a>, S(j))<minD)  minD = dist_to_edge(y<a>,S(j));
    end for
    h_s<a> = min(minC, minD);
```

```
        else if ((type(a)==1) // boundary node
            leftNormal(a)=getLeftNormal(a);
            rightNormal(a)=getRightNormal(a);
            if((leftNormal(a)!=rightNormal(a))||(leftNormal(a)!=rightNormal(a)))
            // corner node
                start=true;
                for (j=0; j<M; j++)
                    if (dist_to_end(y^<a>, S(j))>0)
                        if (start==true)  {minD=dist(y^<a>, S(j)); start=false;}
                        else
                            if (dist_to_edge(y^<a>, S(j))<minD)
                                minD=dist_to_edge(y^<a>, S(j));
                            end if
                        end if
                    end if
                end for //end of for loop on j
                h_s^<a> = min(minC, minD);
            else    // for smooth boundary nodes
                start=true;
                for (j=0; j<M; j++)
                    if (y^<a> is on S(j))
                        dist1 = dist(y^<a>, (x_s(j), y_s(j)));
                        dist2 = dist(y^<a>, (x_e(j), y_e(j)));
                        temp = min(dist1, dist2)
                    else  temp = dist_to_edge(y^<a>, S(j));
                    end if
                    if (start==true)  {minD=temp; start=false;}
                    else
                        if (temp<minD)  minD=temp;
                        end if
                    end if
                end for
                h_s^<a> =min(minC, minD); // set the sub-domain radius
            end if
        end for
    // End of Algorithm 5.4.3
```

Algorithm 5.4.4: Determine the support domain radius for each node.

Given: $\{ y^{<a>}, a=0,1,...N-1\}$ where $y^{<a>}$ represents the location of node a and N is the number of nodes; sub-domain radii $\{ h_s^{<a>}, a=0,1,\ldots,N-1\}$ from algorithm 3.1; safety number N_s; Basis, the current choice of basis (linear, quadratic, or cubic).

To find: support domain radii $\{lw^{<a>}, a=0,1,\ldots N-1\}$

```
/*
```

Note: This algorithm assumes that the following functions are available:
***dist($y^{<a>}$, $y^{<b>}$)** : Returns distance between nodes $y^{<a>}$ and $y^{<b>}$.*
**/*

Step 1: for (a=0; a<N; a++)
lw$^{<a>}$ = $h_s^{<a>}$; *//Initial estimate for* lw$^{<a>}$
close_nodes(a) = -1;
//store the index of nodes in the domain of definition of $y^{<a>}$
dist_close(a)=-1;
//store the distance between the concerned node and other nodes in domain //of definition
far_nodes(a) = -1;
//store the index of nodes outside the domain of definition
dist_far(a)=-1;
//store the distance between the concerned node and other nodes outside //domain of definition
end for *// end of for loop on a*

Step 2: for (a=0; a<N; a++)
if (Basis == linear) N_1=3+N_s;
else if (Basis == quadratic) N_1=6+N_s;
else if (Basis == cubic) N_1=10+N_s;
end if *//end of if-else block on monomial basis*

Step 3: N_2=0; j=0; k=0;
for (b=0; b<N; b++)
if (a!==b)
if (dist($y^{<a>}$, $y^{<b>}$)<lw$^{<b>}$) *// $y^{<b>}$ is in domain of definition of $y^{<a>}$*
N_2=N_2+1; close_nodes(j)=b;
dist_close(j)=dist($y^{<a>}$, $y^{<b>}$); j=j+1;
else *// $y^{<b>}$ is outside domain of definition of $y^{<a>}$*
far_nodes(k)=b; dist_far(k)= dist($y^{<a>}$, $y^{<b>}$);
k=k+1;
end if
end if *//end of if-else block on dist($y^{<a>}$, $y^{<b>}$)*
end for *// end of for loop on b*
J = j; *//the number of nodes in the domain of definition of* $y^{<a>}$
K = k; *//the number of nodes outside the domain of definition of* $y^{<a>}$
close_nodes=sort(close_nodes, dist_close, J);
//rearrange the J first index of close_nodes according to the dist_close
far_nodes=sort(far_nodes,dist_far, K);
//rearrange the K first index of far_nodes according to the dist_far
if (N_1==N_2) break;
else if (N_1>N_2)
for (m=0; m<N_1-N_2; m++)
lw$^{<far_nodes(m)>}$ = 1.1* lw$^{<far_nodes(m)>}$;
//increase the support domain radius by 10% for the nearest N1-N2 //nodes(except the nodes in the domain of definition).
end for
go to Step 3.

```
            else
              for (m=0; m<N1-N2; m++)
                  lw^<close_nodes(K-1-m)> = 0.9* lw^<close_nodes(K-1-m)>;
                        //decrease the support domain radius by 10% for the furthest N2-
                  N1 //nodes in the domain of definition.
              end for
              go to Step 3.
            end if
          end for
// End of Algorithm 5.4.4
```

After we automatically set the sub-domain and support domain for each node using Algorithm 5.4.3 and Algorithm 5.4.4, the user-supplied information and the computed information (node locations, sub-domain radius, support domain radius, and essential boundary conditions) are provided to the meshless solver.

Our pre-processor can be used with most meshless methods, since the required information is usually the same. The user needs to write a translator to convert the input file generated from our pre-processor to the form that is required by the meshless method being used.

5.5 Post-processor

The post-processor provides a convenient graphical user interface for visualization of numerical solutions obtained from the meshless solver. Generally, the post-processors that are designed for use with finite element packages use the idea of an element to produce contours of the desired field variable. The problem domain is usually traversed element by element, and a coloring scheme for the element under consideration is synthesized, based upon the values of the field variable of interest within that element. This approach works very well when the domain has been discretized into elements, but the discretization that is done in a meshless method has no elements and so this approach is inapplicable. Instead, we have developed two truly element-free methods for generating the desired color contours. One method is purely node-based and results in the generation of coarse color contours using a computationally lean algorithm; the second method is pixel-based and generates fine color contours using a method that is computationally more intensive. It is expected that coarse plotting will be used more in the early stages of an analysis or design project, and fine plotting will be used more in the later stages.

In addition to coarse contour plotting and fine contour plotting, the supported displays include path plotting of field variables, geometry plotting, sub-domain plotting and support domain plotting. Table 5.1 outlines the capability of each of these displays, and a more detailed description is presented later in this section. For coarse contour plotting and fine contour plotting, we have two options for the color space: RGB and GRAY. In addition, the fine contour plotting supports a low color resolution mode that is suitable for printing and a high color resolution mode that is suitable for viewing on the screen. The methods for contour plotting that are detailed below are used for both RGB and GRAY color spaces.

To use the post-processor, two ASCII files are needed, both of which can be generated through the GUI. The first file is used to define the global boundary for fine contour plotting. Table 5.2 shows the information in this file for the example of the infinite plate with a circular hole presented in [19]. This example has one boundary comprised of 5 segments. The second file is the output file generated by the meshless method. The format of this file is shown in Table 5.3 and Table 5.4. Table 5.3 shows material properties and parameters of the meshless method that are written out at the top of the output file. Table 5.4 shows the information written out in the output file for each node; this follows the material properties and parameters section in the file. ***Both these files are automatically generated by our meshless solver, so no user effort is needed to create these files***.

5.5.1 Coarse contour plotting

Once we have the output file from the meshless method, we can do either coarse plotting or fine plotting of field variable contours. Using coarse plotting we can quickly generate rough contours of the field variable. This is very important when the nodal density is very large. For problems with about 2800 nodes, fine contour plotting may take 500 times the CPU time that is needed for coarse contour plotting.

The coarse plotting algorithm that we have developed is a node-based method. The basic idea is to traverse the problem domain node by node. At each node we draw a circle of suitable radius centered at that node, and color it uniformly according to the value of the field variable at that node. Clearly, the circles for different nodes should be non-intersecting to avoid a conflict when deciding the color of a particular circle. We accomplish this by making the radius of the

circle centered at a particular node slightly smaller than half the distance to that node's nearest neighboring node. Algorithm 5.5.1 presents the details of the method used for coarse plotting.

Algorithm 5.5.1: Coarse plotting

Given: The field variable value for every node {value(i), i=0,1,…N-1}, where the number of nodes is N; the color degree N_c; type of display, colorspace where colorspace is GRAY or RGB, and size parameter $C \in [0.1,0.4]$.

To find: the color components Color(i, 0), Color(i, 1), Color(i, 2) at node i and the radius Rad(i) of the circle to be drawn centered at node i, i=0,1,…N-1.

```
/*
For RGB colorspace, Color(i, 0), Color(i, 1), Color(i, 2) represent red, green, and
blue percentages at node i;
For GRAY colorspace, Color(i, 0) represents percentage of white and Color(i, 2) represents percentage of
black, with Color (i,1) unused.
*/
```

Step 1: V_{min} = value(0); V_{max} = value(0);
for (i=0; i<N; i++)} *//get the minimum and maximum value for the solution*
V_{min} =min(value(i), V_{min}); V_{max} =max(value(i), V_{min});
end for

Step 2: step = (V_{max}-V_{min})/N_c;
if (N_c is even) $N_{c1} = N_{c2} = N_c /2$;
else { $N_{c1} = (N_c /2)+1$; $N_{c2} = N_c - N_{c1}$;}
end if

Step 3: for (i=0; i<N; i++) *// determine the color and the radius of the circle centered //at each node*

```
if (value(i)==Vmin) degree=0;
else if (value(i)==Vmax) degree= Nc -1;
else
for (j=0; j< Nc; j++)
if (((Vmin+j*step)<=value(i))&&((Vmin+(j+1)*step)>value(i)))
{degree = j; break;}
end if
end for
end if
if (colorspace==GRAY)
Color(i,0) = (degree*160/( Nc -1))/255;
//in order not to make it difficult to recognize nodes on screen, we use
//160 to calculate the lightest color.
Color(i,1) = 0.0;
Color(i,2) = (255-degree*255/( Nc -1))/255;
else if (colorspace==RGB)
if (degree< Nc1) //from blue to green
Color(i,0) = 0.0
Color(i,1) = degree*255/(Nc1-1);
Color(i,2) = 255-degree*255/( Nc1-1);
else //from green to red
```

```
            Color(i,0) = (degree-Nc1)*255/(Nc1-1);
            Color(i,1) = 255-(degree-Nc1)*255/( Nc1-1);
            Color(i,2) = 0.0;
        end if
    end if
    Rad(i) = C* h_s^<i> ;
  end for
// End of Algorithm 5.5.1
```

5.5.2 Fine contour plotting

As noted earlier, mesh-based post-processors typically work by traversing the mesh element by element and determining the color coding for each element based on the computed value of the field variable of interest; this information is then mapped into the device space of the display terminal to generate the image that the user sees. In the meshless post-processor presented here, we work in the opposite direction. Specifically, we traverse the device (screen) space corresponding to the problem domain pixel by pixel, and for each pixel we compute the point in the problem domain space that it maps to; based on the computed value of the field variable of interest at this point, we set the color of the pixel. This allows us to assign colors to points that lie in between nodes. Further, since we address each pixel individually, we are guaranteed to have the highest possible precision in terms of the color coding. In fact, this is as close as we can come to a true continuous variation in the coloring on a discrete (pixel-based) display device.

The post-processor has been implemented in JAVA, and the pixel-based approach that we use for fine contour plotting makes extensive use of the transformation between pixel coordinates and world coordinates. The transformation is implemented using the following:

$$\begin{Bmatrix} x' \\ y' \\ 1 \end{Bmatrix} = \begin{bmatrix} S & 0 & STx \\ 0 & -S & -STy \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} \tag{17}$$

$$\begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = \begin{bmatrix} S & 0 & STx \\ 0 & -S & -STy \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{Bmatrix} x' \\ y' \\ 1 \end{Bmatrix} \tag{18}$$

where $\begin{Bmatrix} x' \\ y' \\ 1 \end{Bmatrix}$ are the pixel coordinates for point P, and $\begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}$ are the world coordinates for point P.

S is the factor by which coordinates are scaled along both the X and the Y directions; Tx is the distance by which coordinates are translated in the X direction and Ty is the distance by which coordinates are translated in the Y direction. If we know the pixel coordinates for point P, we can get the corresponding world coordinates using Equation 18.

The coefficients S, Tx, and Ty can be easily found once we know our area interest in the problem domain and our display area on the screen. For this purpose, we employ a world coordinate system oxy and pixel coordinate system o'x'y' as shown in Figure 5.7. We assume that a one rectangular area of interest in the world coordinate system oxy maps to a known rectangular area in the pixel coordinate system o'x'y'. We also assume that we want the center C of the rectangle in the world coordinate system to be mapped to the center C' of the rectangle in the pixel coordinate system after transformation. The width W and height H of the rectangular area of interest in the world coordinate system are known from the problem extents or user-specified zoom limits. The width W' and height H' of the display area are found by querying the JAVA component on which the drawing is done. We can get the scaling factor scaleX in the X direction by noting that $\text{scaleX}=\frac{W'}{W}$. Similarly, the scaling factor scaleY in the Y direction is given by $\text{scaleY}=\frac{H'}{H}$. We will select the lower of scaleX and scaleY to be the scaling factor S along both X and Y axis directions. The coordinates of the point C are $(\frac{W}{2},\frac{H}{2})$ in world coordinates, and the coordinates of point C' are $(\frac{W'}{2},\frac{H'}{2})$ in pixel coordinates. After we get the scaling factor S, according to Equation (17), we can calculate Tx and Ty:

$$\text{Tx}=\frac{W'-SW}{2S} \qquad \text{Ty}=\frac{H'+SH}{-2S}.$$

In addition to the pixel-to-world and world-to-pixel coordinates, we need to be able to calculate the value of the field variable at any point in the domain. This is done by reading the

actual displacements, fictitious displacements, stresses, and strains at each node from the output file and applying Equations (13) through (16), depending on the field variable that we are working with.

For fine contour plotting, we support two different kinds of plotting: low color resolution fine plotting and high color resolution fine plotting. For the former, we use only limited levels in the color scale (usually 8), but for the latter we use 512 colors for RGB color space and 256 colors for GRAY color space. The low color resolution plotting is intended for printing purposes because of the higher contrast between colors, while high color resolution plotting is intended for viewing on screen. Algorithm 5.5.2 below describes the details of the pixel-based fine contour generation algorithm.

Algorithm 5.5.2: Fine plotting

Given: The global boundary segments {S(k), k=0,1,…M-1} where M is the number of Segments; the fictitious displacement of each node {fict_dispX(i), fict_dispY(i), i=0,1,…N-1}, where the no. of nodes is N; the location for each node loc(i)=(x(i), y(i)), i=0,1,…N-1; the color degree N_c; type of display, colorspace where colorspace is GRAY or RGB.

To find: the color for each pixel that is located in this problem domain in pixel coordinates.

/ *Existence of the following function is assumed:*
toPixel(x): *Return the pixel coordinate of world space point x (See Equation (17)).*
**/*

Step 1: $x_{min} = x(0)$; $x_{max} = x(0)$; $y_{min} = y(0)$; $y_{max} = y(0)$;
for (i=0; i<N; i++) *//get the minimal and maximal value for x and y*
$x_{min} = \min(x(i), x_{min})$; $x_{max} = \max(x(i), x_{max})$;
$y_{min} = \min(y(i), y_{min})$; $y_{max} = \max(y(i), y_{max})$;
end for

Step 2: $(x'_{min}, y'_{min}, 1) = toPixel((x_{min}, y_{min}, 1))$; *//use Equation (17) to get the values*
$(x'_{max}, y'_{max}, 1) = toPixel((x_{max}, y_{max}, 1))$; *//in pixel coordinates*

Step 3: $Nox = x'_{max} - x'_{min}$; $Noy = y'_{max} - y'_{min}$;
$divX = (x_{max} - x_{min})/Nox$; $divY = (y_{max} - y_{min})/Noy$;
$N_l = 0$;

Step 4: k=0; *//index of the pixel in the problem domain*

for (i=0; i<Noy; i++) *//go through each pixel in the minimum rectangle to see //if this pixel is in the problem domain or not*

$y = y_{min} + divY * i$;
for (j=0; j<Nox; j++)
$x = x_{min} + divX * j$; worldPt=(x, y);
referPt=(x_{min}, y); *//on the left side of the minimum rectangle*
seg=Line(worldPt, referPt); *//connecting the referPt and worldPt*
insection=0; *// the number of insection between seg and the boundary*

onsegment=false;
for (k=0; k<M; k++) *//check the relationship between seg and boundary*
if (worldPt is on S(k)) {onsegment=true; break;}
end if
if (insection(S(k), seg)) insection= insection +1;
end if
end for
if ((insection==odd)||(onsegment==true))
// if insection is odd, the worldpt is in the problem domain
$N_1 = N_1+1$;
(N_x, domainofDefinition)=getDomainOfDefinition(worldPt);
//N_x is the no. of nodes in the domain of definition of worldPt
//domainofDefinition is the array of index of nodes in the domain
//of definition
out=getTheSolutionValue(Nx, domainofDefinition, worldPt);
// refer to Equations.13-16
PtInDomain(k)=worldPt; *//Save the point in an array*
PtInDomainValue(k)=out; *//Save corresponding value of varia*ble
k=k+1;
end if
end for *//end of for loop on j*
end for *//end of for loop on i*

Step 5: V_{min}=min{PtInDomainValue(i), i=0,…,N_1-1};
V_{max}=max{PtInDomainValue(i), i=0,…, N_1-1};
step=(V_{max}-V_{min})/N_c;
if (N_c is even) N_{c1}=N_{c2}=N_c /2;
else { N_{c1}= N_c /2+1; N_{c2}=N_c-N_{c1};}
end if *//end of if-else block on N_c*

Step 6: for (i=0; i<N_1; i++) *// determine the color for each pixel inside the problem //domain*
value= PtInDomainValue(i);
if (value==V_{min}) degree=0;
else if (value==V_{max}) degree=N_c-1;
else
for (j=0; j<N_c; j++)
if ((V_{min}+j*step)<=value)&&((V_{min}+(j+1)*step)>value)
{degree=j; break;}
end if
end for
end if
if (colorspace==GRAY)
Color(i,0)=(degree*160/(N_c -1))/255;
//in order not to make it difficult to recognize nodes on screen, we use
//160 to calculate the lightest color.
Color(i,1)=0.0;
Color(i,2)=(255-degree*255/(N_c -1))/255;

```
            else if (colorspace==RGB)
        if (degree< Nc1) //from blue to green
            Color(i,0)=0.0
            Color(i,1)=degree*255/(Nc1-1);
            Color(i,2)=255-degree*255/( Nc1-1);
        else //from green to red
            Color(i,0)=(degree-Nc1)*255/(Nc1-1);
            Color(i,1)=255-(degree-Nc1)*255/( Nc1-1);
            Color(i,2)=0.0;
        end if
      end if
    end for
// End of Algorithm 5.5.2
```

5.5.3 Geometry plotting

Using geometry plotting, we can draw the undeformed and deformed geometry of the problem domain using the data provided in the second file describe in this chapter. For the undeformed geometry, we will draw the initial position for every node along with the boundaries given by the user. For deformed geometry, we just the draw the final position for each node. In both case, the size to which each node is drawn is user controllable.

5.5.4 Path plotting

Using this function, we can see how field variables of interest are distributed along a specified straight line in the problem domain. In addition to selecting the field variable to be plotted, the only data that the user needs to provide are the coordinates of the two end points of this line and the number of divisions to be used for plotting along this line.

5.5.5 Sub-domain plotting and support domain plotting

The sub-domain radius and support domain radius for each node are important parameters. Using these functionalities, we can directly visualize the sub-domains and support domains that are defined. For sub-domain distribution, we can know if the sub-domain of any node exceeds the problem domain or not. For support domain distribution, we can know if the support domain radius is suitable or not. A zoom-in/zoom-out facility is provided to allow closer inspection of sub-domains and support domains in any region of interest.

Our post-processor can generally be used with other meshless methods as long as a data translator is provided to convert the format of the output file into the format our meshless method

uses. Since all meshless methods use fictitious displacement to construct the system equations, and all the files required by the post-processor are ASCII, this is easy to do.

5.6 Example

This section presents in detail an example to illustrate the capabilities of the pre-processor and the post-processor to produce graphical representations of the results obtained using the meshless method. The example is an infinite plate with a circular hole.

The Figure 5.8 shows the interface of the software package that was developed using JAVA. This software gives us the flexibility of changing the color degree and deformation factor and switching the color space between RGB and GRAY. Changing the deformation factor will exaggerate the actual deformation when the actual deformation is very small.

The test example is an infinite plate with a circular hole in the center subjected to a uniform remote tension σ in x_1 direction which is used for testing in [19]. The exact solutions for the stresses are given in [25], and agree well with our results.

Because of symmetry, only the upper right quadrant of the plate was modeled. The model geometry has 336 nodes with more nodes concentrated around the hole. The material properties used in the calculation are E=1000 MPA, ν=0.3 with plane strain condition, and $\sigma = 1$ MPA. The essential boundary conditions were applied on the right, the upper, the left and the bottom edges. On the remaining edge (the inner circle), the natural boundary conditions (free traction) were applied.

To generate the input file, we need to use the GUI to enter the geometric model, as well as definition of material, boundary conditions, and other parameters for the meshless method. Then by using our pre-processor, we can automatically generate the input data for the meshless solver, from which we can get the output file needed for our post-processor.

Contour plotting includes coarse plotting, low color resolution contour plotting and high color resolution contour plotting. For each function, we can plot the contour plots for displacements (u_1, u_2), stress (σ_{11}, σ_{22}, and σ_{12}), strain (ε_{11}, ε_{22}, and ε_{12}), principle stress ($\sigma^{(1)}, \sigma^{(2)}$), and principle strain ($\varepsilon^{(1)}, \varepsilon^{(2)}$).

5.6.1 Coarse contour plotting

Figure 5.9 gives the plots for several field solutions for visualizing field variable contours.

5.6.2 Low color resolution fine plotting

Given a certain number of color degree, using this function we can draw the contour for the whole problem domain. Low color resolution contour plots for RGB color space are presented in Figure 5.10a and Figure 5.10b; low color resolution contour plots for GRAY color space are presented in Figure 5.10c and Figure 5.10d.

5.6.3 High color resolution fine plotting

This function also draws the contour for the whole problem domain. High color resolution contour plots for RGB color space are presented in Figure 5.11a and Figure 5.11b; high color resolution contour plots for GRAY color space are presented in Figure 5.11c and Figure 5.11d. When we compare Figure 5.10 and Figure 5.11, we see that Figure 5.10 is easier to interpret better than Figure 5.11 for printing because of the high color contrast between colors. However, the plot shown in Figure 5.11 is better than that in Figure 5.9 for viewing on screen.

5.6.4 Path plotting

Given a starting point and ending point along with the division number, using this function we can get the distribution of field solutions along this straight line. The path plot for σ_{11} is presented in Figure 5.12a.

5.6.5 Sub-domain plotting

Using this function, we can draw the sub-domain for each node. This can give us a better understanding of the sub-domain size for all nodes and how the sub-domain of one node is coupled with those of others. Figure 5.12b gives the sub-domain plot for this example.

5.6.6 Support domain plotting

We draw the support domain for each node to visualize its geometry. For any node, we can see which other nodes will have an effect on this particular node and the locations of these nodes. A support domain plot is presented in Figure 5.12c.

5.6.7 Geometry Plotting

We can draw the undeformed and deformed geometry of the problem domain for better understanding of the geometry of the problem. Geometry plots are presented in Figure 5.13.

In this example, the coarse plotting can be done within one second because the field solution values are either available in the postprocessor.txt (for displacements, stresses and strains) or can be gotten very quickly (for principle stresses and principle strains). Low color resolution fine contour plotting and high color resolution fine contour plotting can be done about 9 minutes approximately for the node distribution that is shown.

5.7 Concluding Remarks

In recent years, meshless methods have been developed as an alternative numerical approach to eliminate the known drawbacks in the well-known finite element method, meshless methods do not require a mesh to discretize the problem domain, and the approximate solution is constructed entirely based on a set of scattered nodes. To use a meshless method, the user should generate an input file in which the nodes information is included. This paper presents a quadrilateral-based pre-processor to define the data and discretization scheme for meshless analysis. To use the pre-processor, the user needs to use the graphic user interfaces to enter the geometric model, as well as definition of material, boundary conditions, and other parameters for the meshless method

To better understand the results obtained from meshless methods, the user may need post-processing support for analyzing and visualizing the results. In this paper, we describe a post-processor that is truly meshless, and can therefore be used directly with meshless solvers. In the meshless post-processor presented here, we traverse the device space pixel by pixel and for each pixel, compute the point in the problem domain space that it maps to; based on the computed value of the field variable of interest at this point, we set the color of the pixel. By working from device space to domain space instead of the other way around, we are able to completely eliminate the need for a mesh to support post processing.

The graphic user interfaces for pre-processor are very friendly and easy to use, and the pre-processor and post-processor can be made to work with any meshless solver. One extensive examples is presented to demonstrate the capabilities of the pre-processor and the post-processor.

The results show that the pre-processor and the post-processor are effective and the latter is computationally viable visualization tool for meshless methods.

5.8 References

[1] Mukherjee, Y.X. and Mukherjee, S. Boundary node method for potential problems. International Journal for Numerical Methods in Engineering 1997; 40: 797-815.

[2] Chati, M.K. and Mukherjee, S. The boundary node method for 3D problems in potential theory. International Journal for Numerical Methods in Engineering 2000; 47: 1523-1547.

[3] Gu, Y.T. and Liu, G.R. A boundary point interpolation method for stress analysis of solids. Computational Mechanics 2002; 28: 47-54.

[4] Nayroles, B., Touzot and G., Villon, P. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational Mechanics 1992; 10: 307-318.

[5] Belytschko, T., Lu, Y.Y. and Gu, L. Element free Galerkin method. International Journal for Numerical Methods in Engineering 1994; 37: 229-256.

[6] Liu, W.K. and Chen. Y. Wavelet and multiple scale reproducing kernel methods. International Journal for Numerical Methods in Fluids 1995; 21: 901-931.

[7] Aluru, N.R. A reproducing kernel particle method for meshless analysis of microelectromechanical systems. Computational Mechanics 1999; 23: 324-307.

[8] Chen, J.S., Pan, C., and Wu, C.T. A Lagrangian reproducing kernel particle method for metal forming analysis. Computational Mechanics 1998; 22: 289-338.

[9] Liu, G.R. and Gu, Y.T. A point interpolation method for 2D solid. International Journal for Numerical Methods in Engineering 2001; 50: 937-951.

[10] Liu, G.R. and Yan, L. A study on numerical integration in element-free methods. Proceeding of 4th Asia-Pacific Conference on Computational Mechanics, Singapore.1999: 979-984.

[11] Atluri, S.N. and Zhu, T. A new meshless Local Petrov-Galerkin approach in computational mechanics. Computational Mechanics 1998; 22: 117-127.

[12] Atluri, S.N. and Zhu, T. New concepts in meshless methods. International Journal for Numerical Methods in Engineering 2000; 47: 537-556.

[13] Atluri, S.N. and Zhu, T. The Meshless Local Petrov-Galerkin approach for solving problems in elasto-statics. Computational Mechanics 2000; 225: 169-179.

[14] Liu, G.R. and Gu, Y.T. Meshless local Petrov-Galerkin (MLPG) method in combination with finite element and boundary element approaches. Computational Mechanics 2000; 26(6):536-546.

[15] Zhu T., Zhang J-D., and Atluri S.N. A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Computational Mechanics 1998; 21: 223-235.

[16] Atluri S.N., Sladeck J., Sladeck V., and Zhu T. The local boundary integral equation (LBIE) and its meshless implementation for linear elasticity. Computational Mechanics 2000; 25: 180-198.

[17] Sladek, J., Sladeck, V., and Van Keer, R. Meshless local boundary integral equation for 2D elastodynamic problems. International Journal for Numerical Methods in Engineering 2003: 235-249.

[18] Long, S. and Zhang, Q. Analysis of thin plates by the local boundary integral equation (LBIE) method. Engineering Analysis with Boundary Elements 2002; 26: 707-718.

[19] Bodin, A., Ma, J., Xin, X.J. and Krishnaswami, P. A Meshless Integral Method Based on Regularized Boundary Integral Equation. Computational Methods in Applied Mechanics and Engineering. (In press).

[20] Liu, G.R. and Gu, Y.T. A local point interpolation method for 2D solid. International J. Struct. Eng. Mech. 2001; 11(2):221-236.

[21] ANSYS 5.6 Documentation, ANSYS Incorporated, 1999.

[22] The MacNeal-Schwendler Corporation, MSC/PATRAN user's manual. Los Angeles, CA: The MacNeal-Schwendler Corporation, 1996.

[23] James, Mark. A Finite Element Model of Fluid Flow in Jointed Rock. Master thesis. Kansas State University. 1988.

[24] MFree2D Software. http://www.nus.edu.sg/ACES/software/meshless2D.htm. 1999.

[25] Timoshenko, S.P., Goodier, J.N. *Theory of elasticity*. McGraw-Hill Book Company, 1970.

[26] Stroud, A.H. and Secrest, D. *Gaussian Quadrature Formulas*. Prentice-Hall, 1966.

[27] Lu, YY, Belytschko T, and Gu, L. A new implementation of the element free Galerkin method. Computer Methods in Applied Mechanics and Engineering 1994; 113: 397-414.

[28] Gavete, L., Benito, J.J., Falcon S. and Ruiz A. Implementation of essential boundary conditions in a meshless method. *Commun. Numer. Meth. Engng.* 2000; 16: 409-421.

[29] Arnold DN, Brezzi F, Cockburn B, Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. SIMA J. Numer. Anal. 2002; 39: 1749-1779.

[30] Hegen D., Element-free Galerkin methods in combination with finite element approaches. Computer Methods in Applied Mechanics and Engineering 1996; 19: 120-135.

[31] Gunther, F. C. and Liu, W. K. Implementation of boundary conditions for meshless methods. Computer Methods in Applied Mechanics and Engineering 1998; 163: 205-230.

[32] Duarte CAM and Oden JT, An h-p adaptive method using clouds, Comput. Methods Appl. Mech. Engrg. 1996; 139: 237-262.

[33] Lu YY, Belytschko T, Tabbara M. Element-free Galerkin method for wave propagation and dynamic fracture. Computer Methods in Applied Mechanics and Engineering 1995; 126: 131-153.

[34] Zhang X, Liu X, Lu MW and Chen Y, Imposition of essential boundary conditions by displacement constraint equations in meshless methods. Communications in Numerical Methods in Engineering 2001; 17: 165-178.

[35] Wagner, GJ and Liu WK, Application of essential boundary conditions in mesh-free methods: a corrected collocation method. International Journal for Numerical Methods in Engineering 2000; 47: 1367-1379.

[36] Zhu, T. and Atluri, S.N. A modified collocation method and a penalty foumulation for enforcing the essential boundary conditions in the element free Galerkin method. Comput. Mech. 1998; 21(3): 211-222.

[37] Wu CC and Plesha ME, Essential boundary condition enforcement in meshless methods: boundary flux collocation method, Int. J. Numer. Meth. Engng. 2002; 53: 499-514.

Figure 5.1 Schematic diagram illustrating the meaning of local sub-domain and support domain

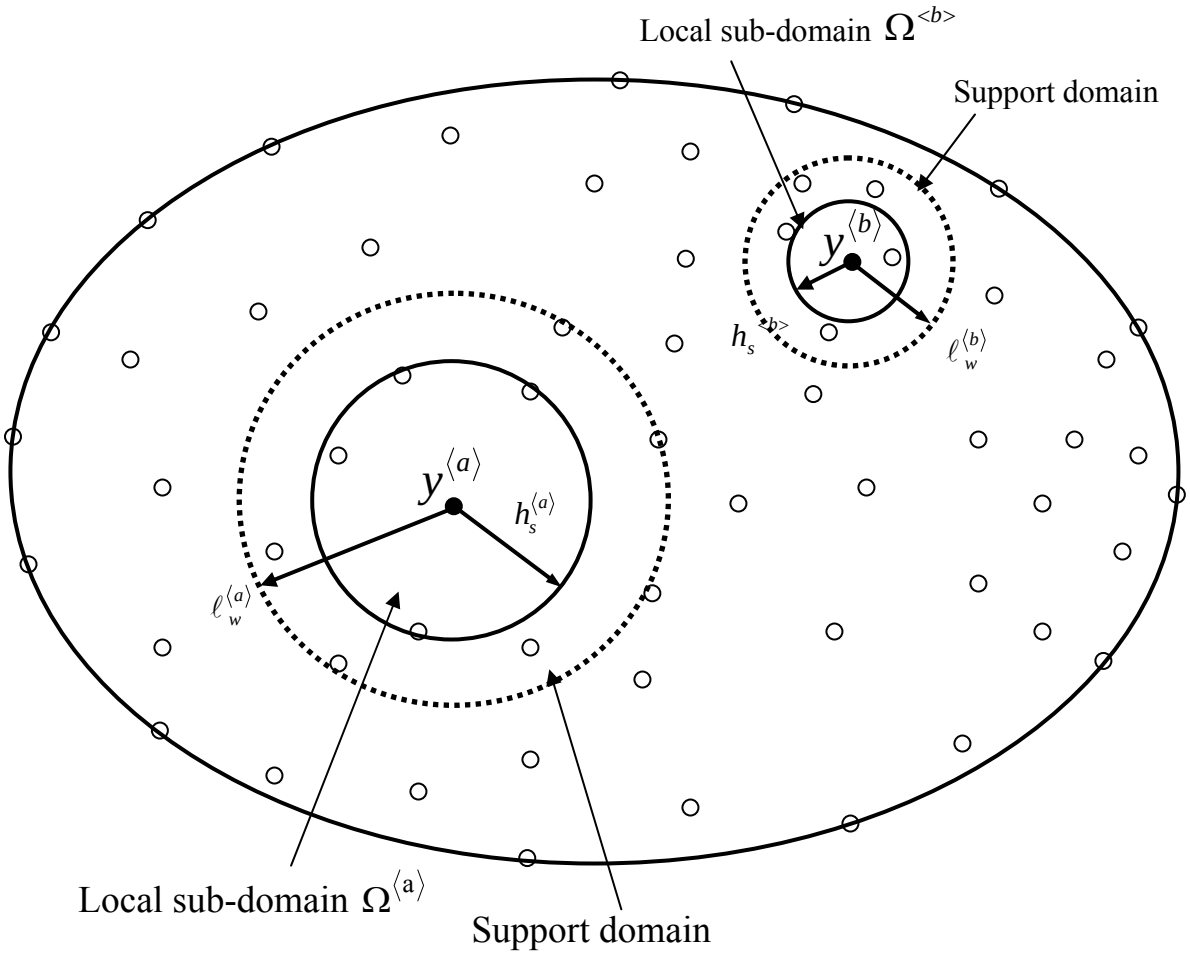


Figure 5.2 Schematic diagram showing the domain of influence for a node $\mathbf{y}^{<a>}$ and the domain of definition for a point $\mathbf{x}$

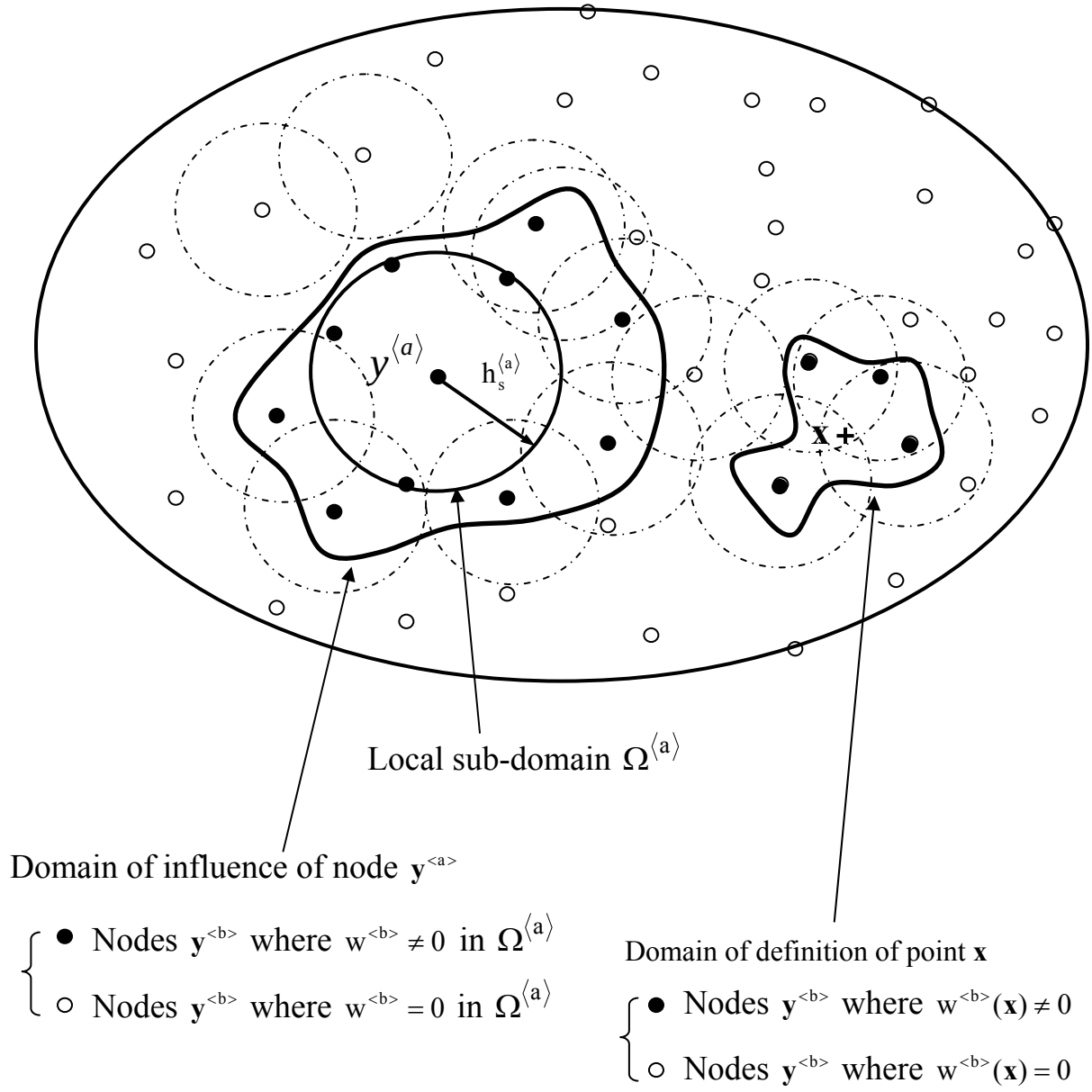
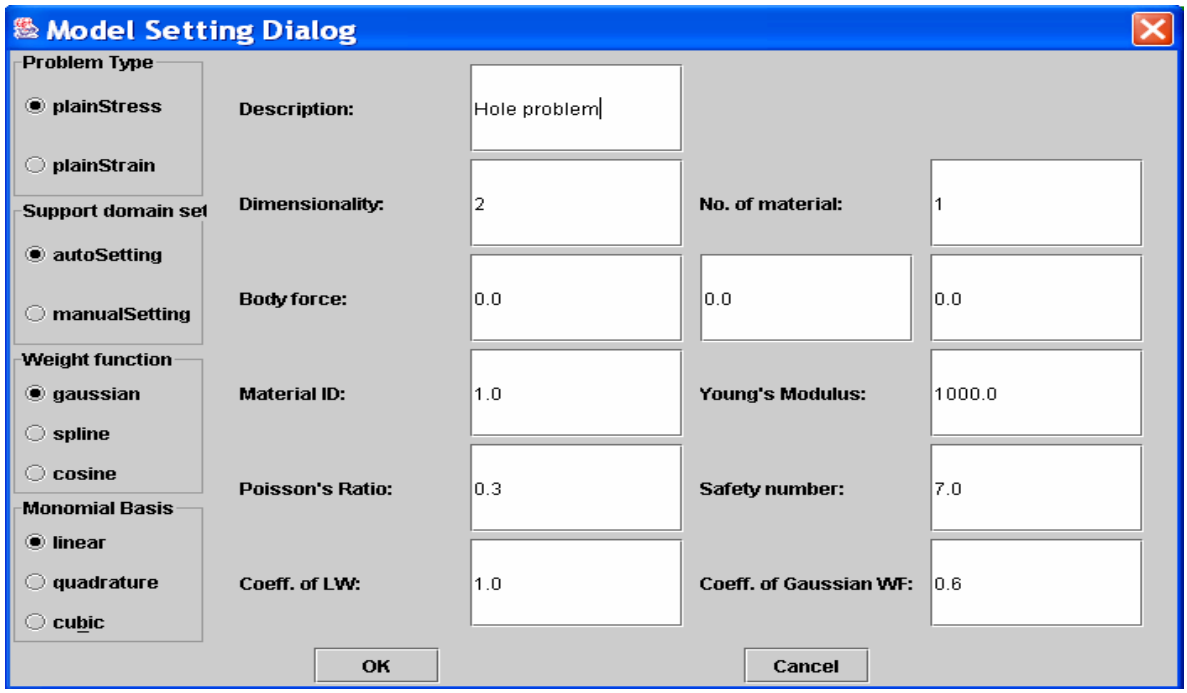
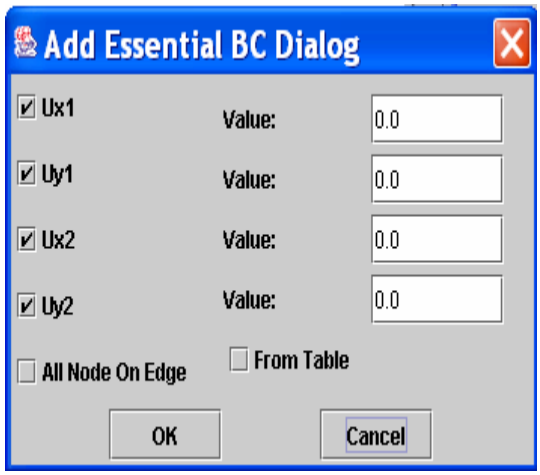


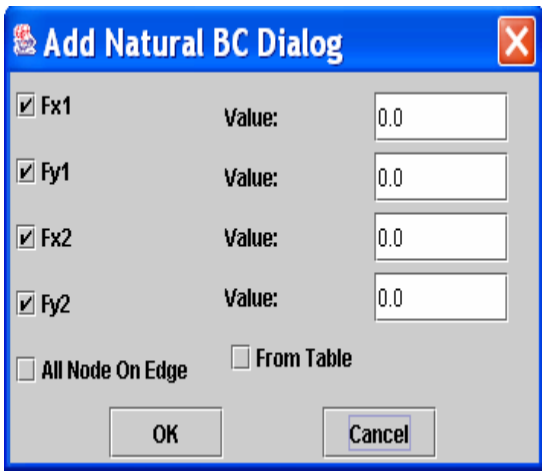
Figure 5.3 Interfaces for setting model information, essential boundary condition, and natural boundary



(a)



(b)



(c)

Figure 5.4 Internal nodal refinement

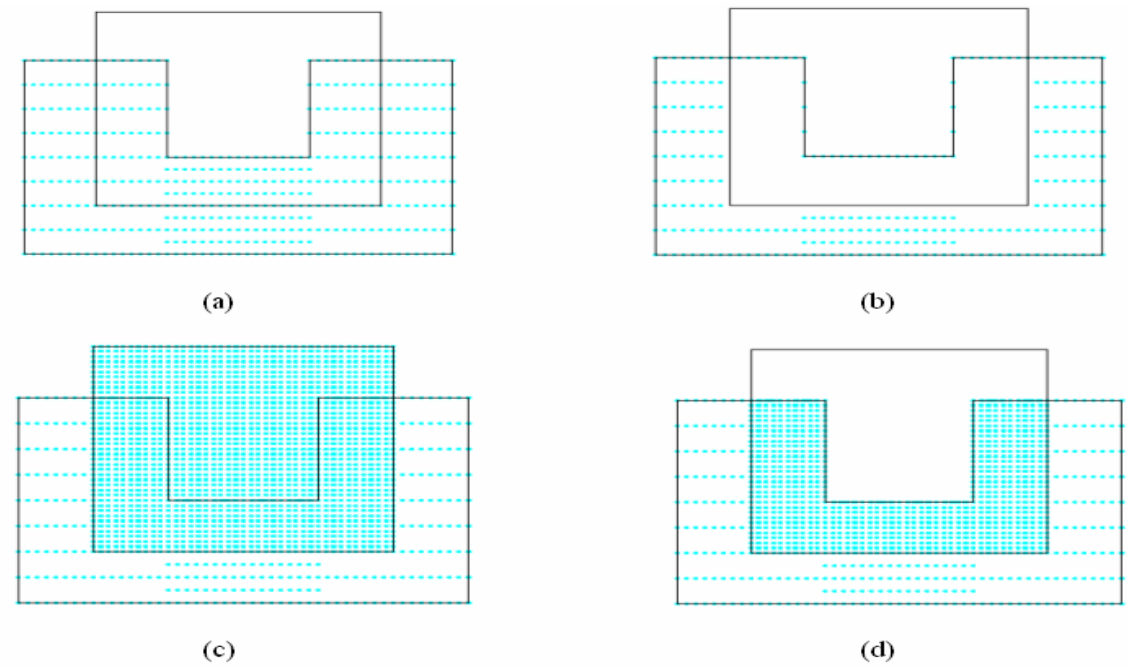


Figure 5.5 Nodal refinement on boundary

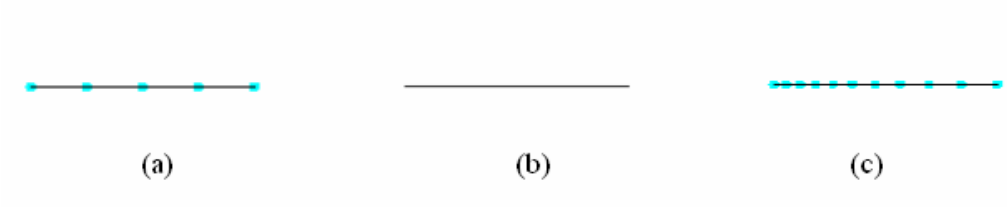
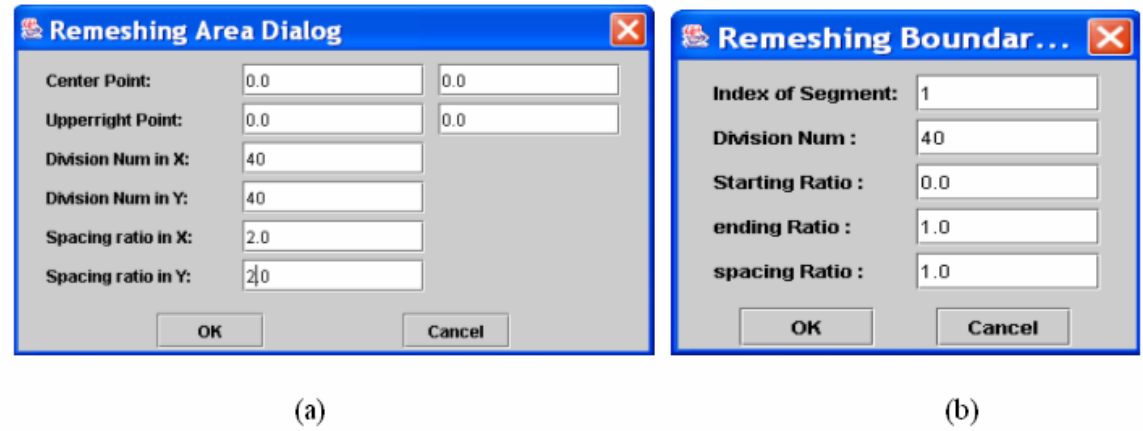


Figure 5.6 Nodal refinement dialogs



(a) (b)

Figure 5.7 Schematic diagram showing how we get the parameters S, Tx, and Ty for the transformation between pixel coordinates and world coordinates

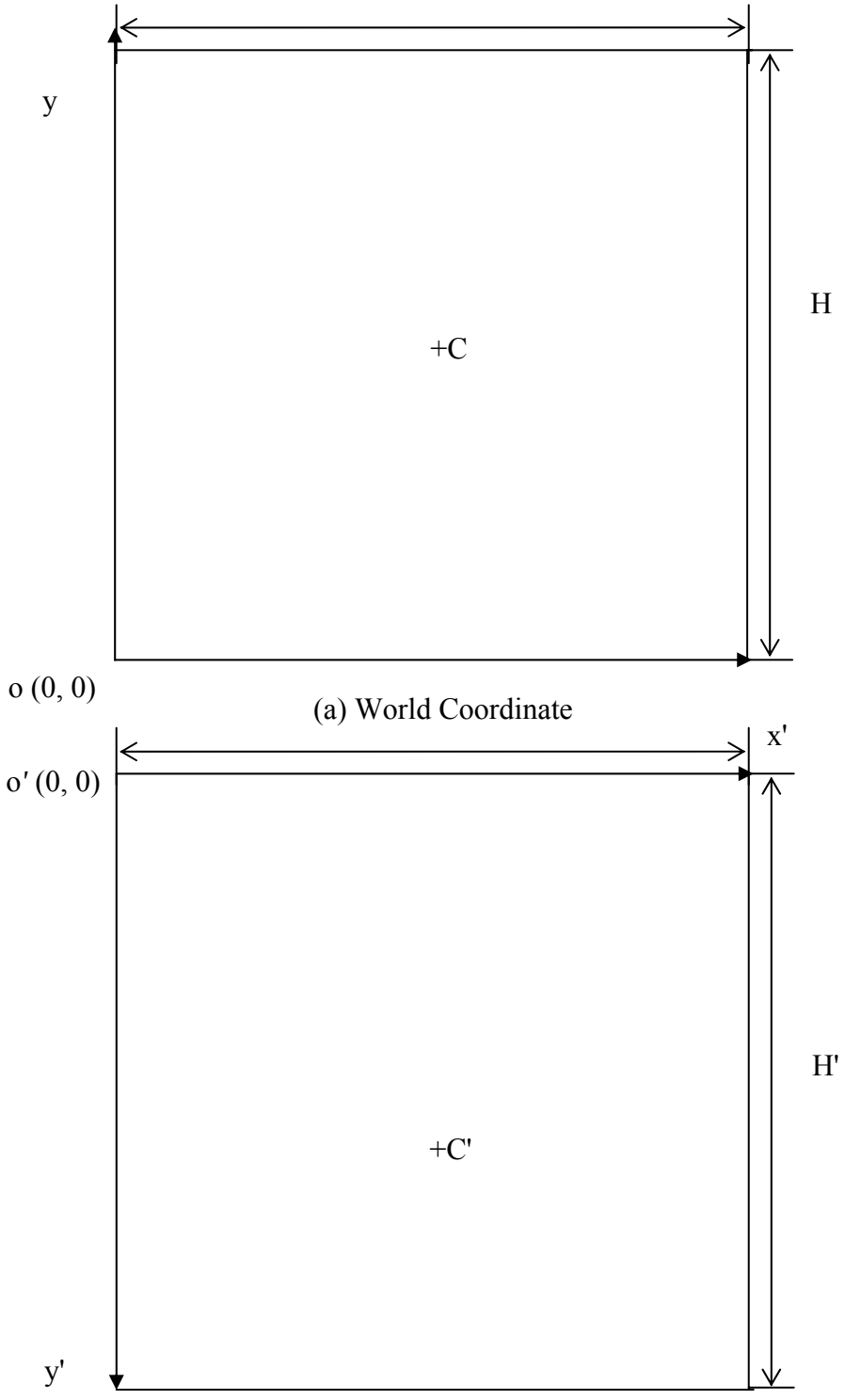


Figure 5.8 Interface of the software

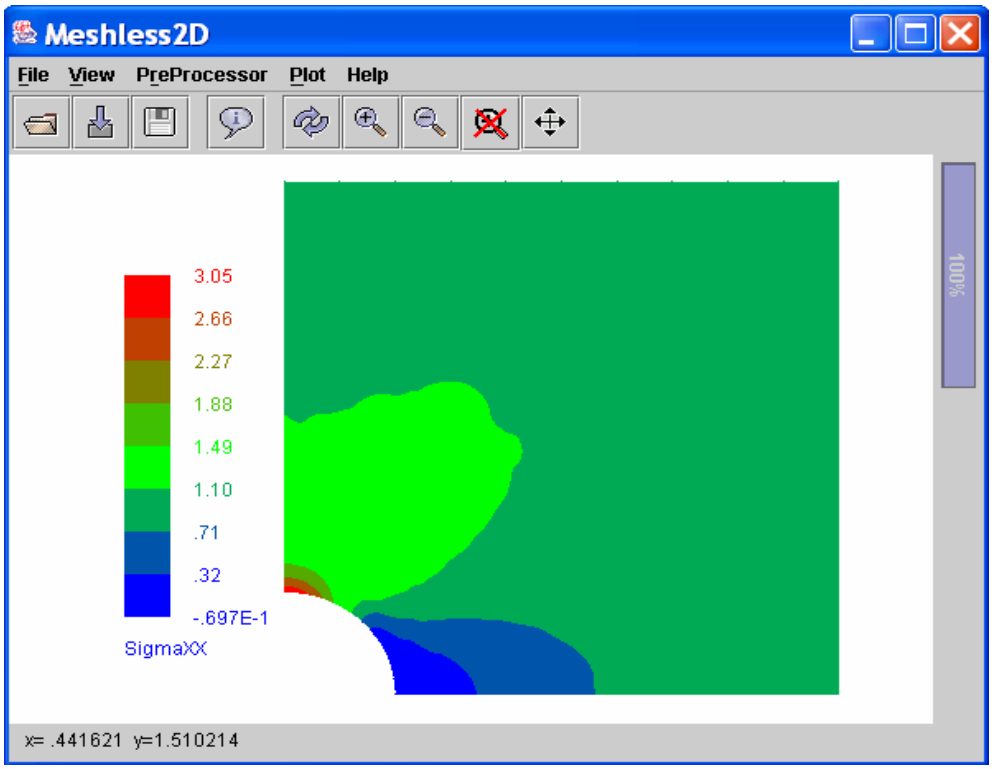


Figure 5.9 Coarse contour plots for infinite plate with a circular hole. (a) (b) RGB color space. (c) (d) GRAY color space plate

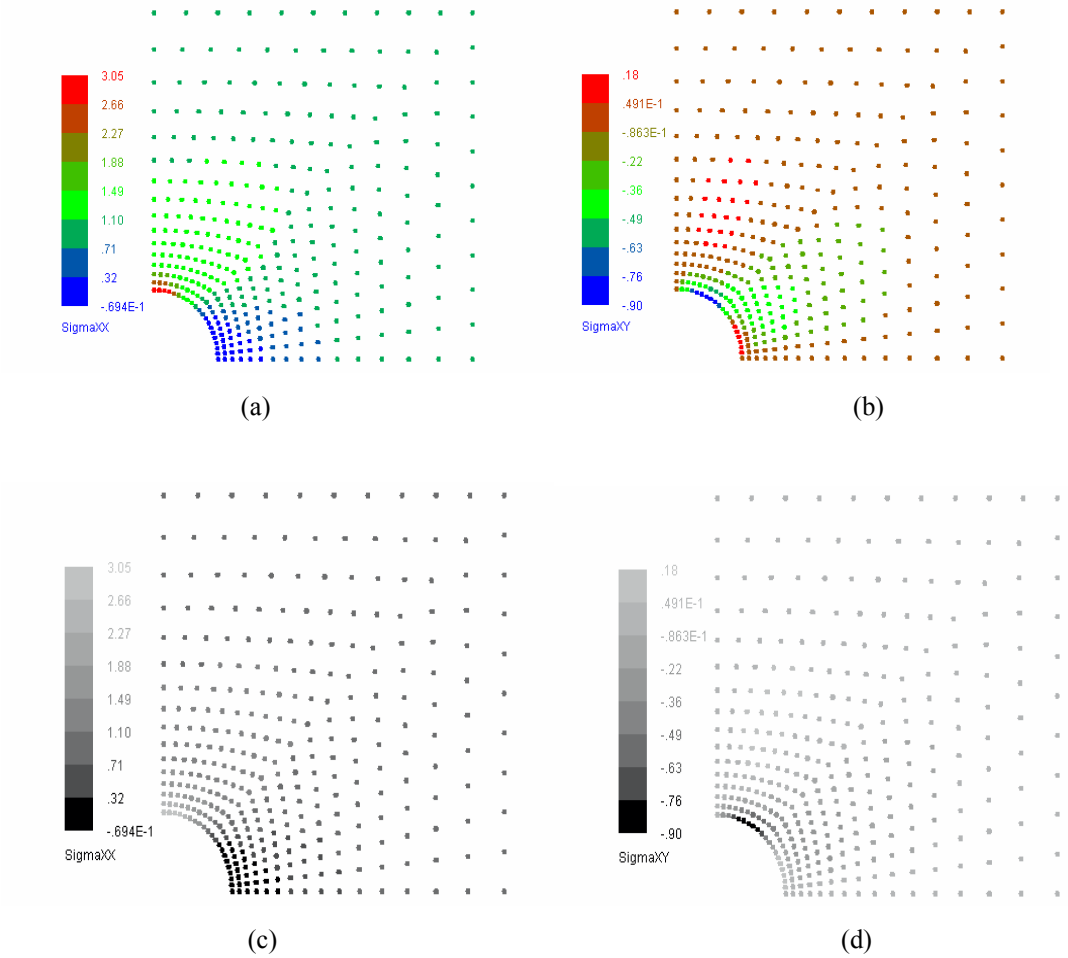
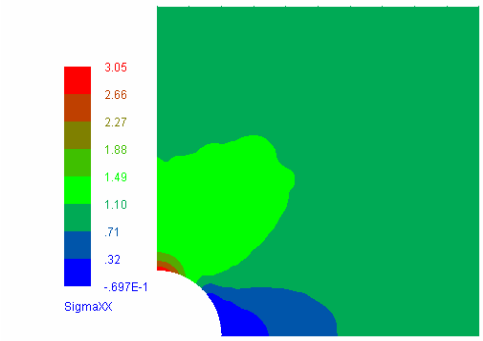
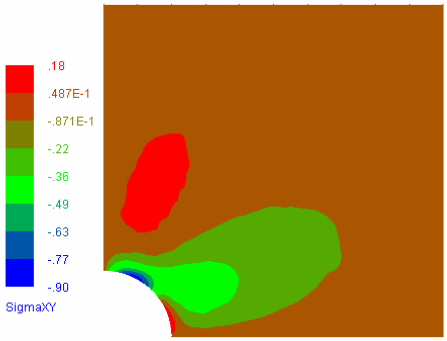


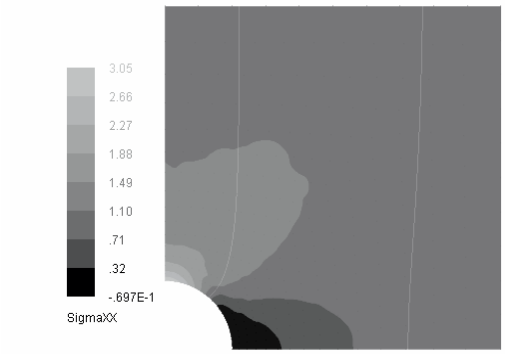
Figure 5.10 Low color resolution contour plots for infinite plate with a circular hole



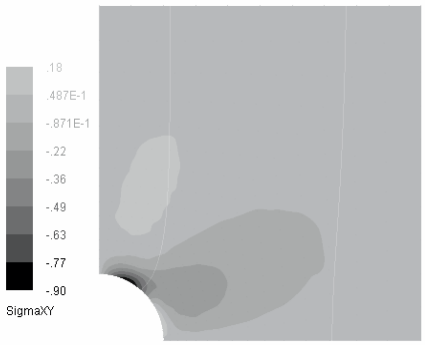
(a)



(b)



(c)



(d)

Figure 5.11 High color resolution contour plots for infinite plate with a circular hole

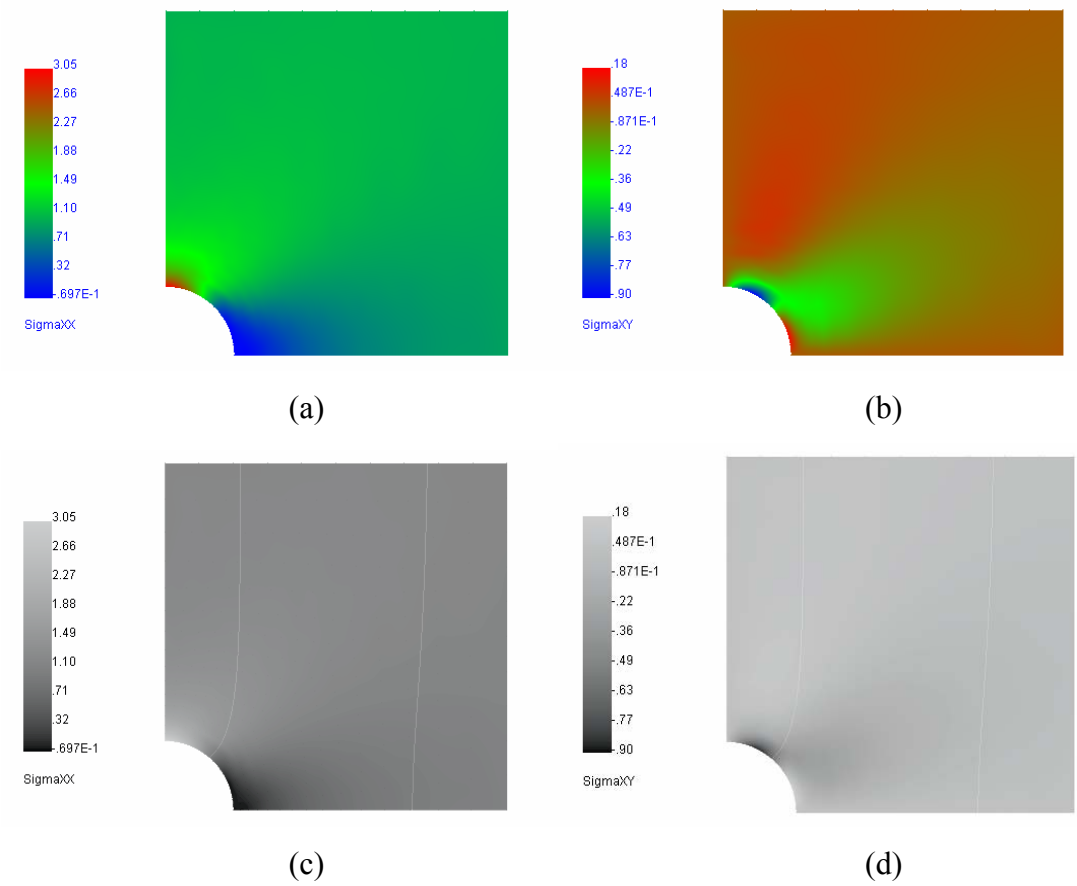
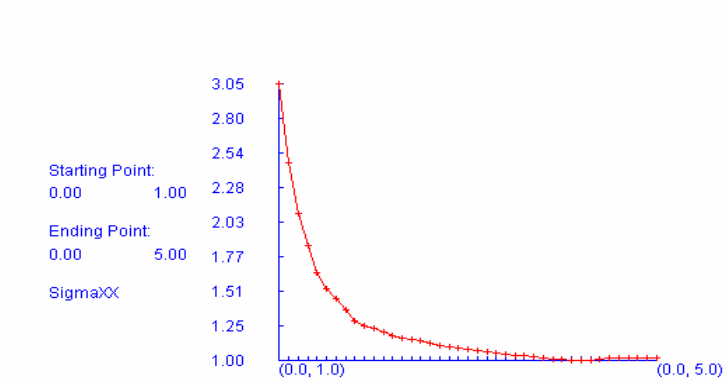
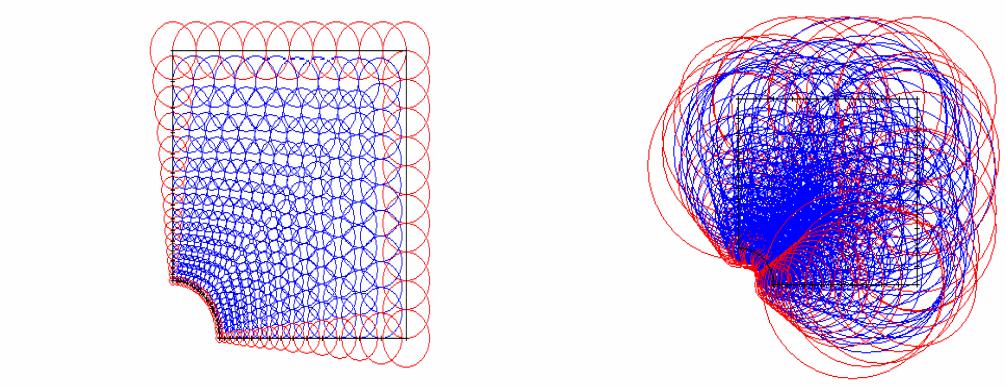


Figure 5.12 Infinite plate with a circular hole. (a) Path plot. (b) Sub-domain plot. (c) Support domain plot



(a)



(b) (c)

Figure 5.13 Geometric plots for infinite plate with a circular hole

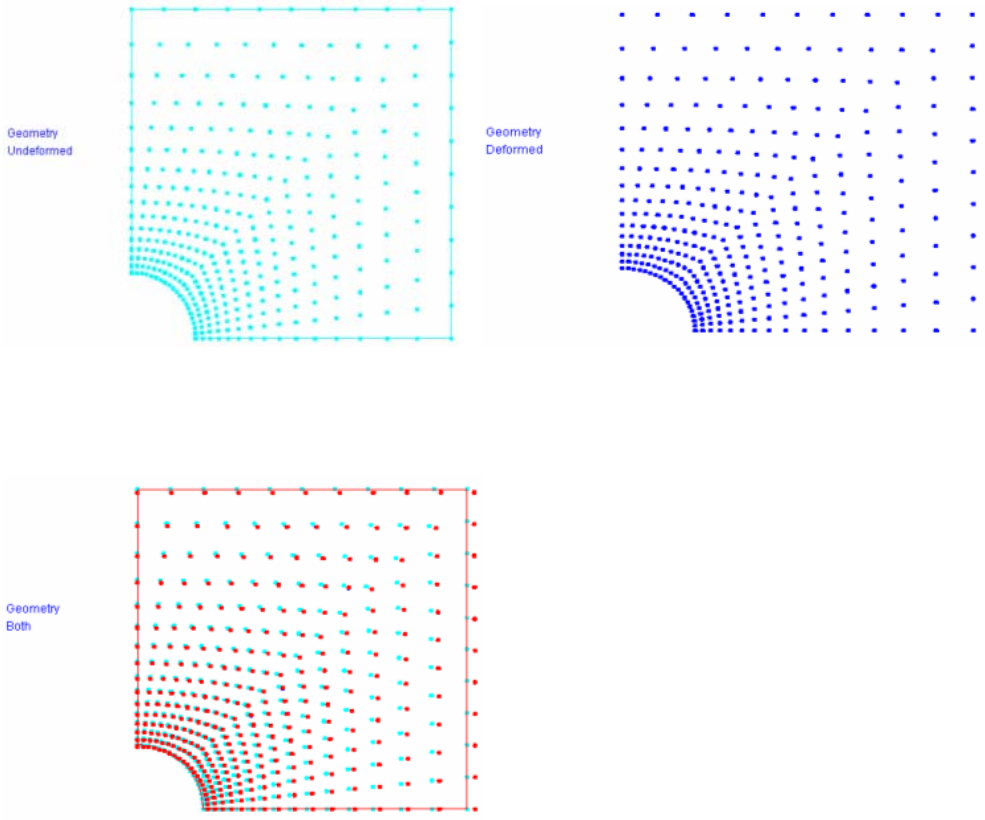


Table 5.1 The functionalities the post-processor has

Content	Explanation
Coarse plotting	draw the contour plot only at known nodes
Low color resolution contour plotting	draw the contour plot at each point using limited levels in the color scale
High color resolution contour plotting	draw the contour plot at each point using high levels in the color scale
Path plotting	draw the distribution of concerned solution along a given path
Geometry plotting	draw the undeformed or deformed geometry or both
sub-domain plotting	draw the distribution of sub-domain for each node
support domain plotting	draw the distribution of support domain for each node

Table 5.2 The complete information the first file contains in post-processor for hole example

Content						Explanation
1						number of boundaries
5						number of segments of the first boundary
LINE						Type of 1st segment of 1st boundary
0.0	1.0	0.0	5.0			start point and end point; both u_1 and u_2 are specified
LINE						Type of 2nd segment of 1st boundary
0.0	5.0	5.0	5.0			start point and end point; both u_1 and u_2 are specified
LINE						Type of 3rd segment of 1st boundary
5.0	5.0	5.0	0.0			start point and end point; both u_1 and u_2 are specified
LINE						Type of 4th segment of 1st boundary
5.0	0.0	1.0	0.0			start point and end point; both u_1 and u_2 are specified
ARC						Type of 5th segment of 1st boundary
0	0	1	0	0	1	center, start point and end point; neither u_1 nor u_2 is specified

Table 5.3 Material properties and parameters of the meshless method for hole example

Content	Value	Explanation
No	336	number of nodes
E	1000	Young's modulus
Nu	0.3	Poisson's ratio
PlaneType	0	0: plane strain; 1: plane stress
FunKind	1	type of weight function; (0: Gaussian; 1: spline; 2:cosine)
basis	1	type of monomial basis; (1: linear; 2:quadratic; 3:cubic)

Table 5.4 The information written out in the output file by meshless solver for each node

Content	Value	Explanation
X	0.0	X coordinate of this node
Y	5.0	Y coordinate of this node
UhX	9.26E-6	fictitious displacement value of this node in X
UhY	-0.0022	fictitious displacement value of this node in Y
Ux	0.0	actual displacement value of this node in X
Uy	-0.0022	actual displacement value of this node in Y
SigXX	1.0194	stress value of σ_{11}
SigYY	-0.0194	stress value of σ_{22}
SigXY	0.01	stress value of σ_{12}
Exx	0.00094	strain value of ε_{11}
Eyy	-0.00012	strain value of ε_{22}
Exy	-0.00039	strain value of ε_{12}
Sub-domain	0.5	radius of the sub-domain for this node
Support domain	1.65	radius of the support domain
Type of node	1	1: boundary node; 0: interior node

CHAPTER 6 - Over Summary and Conclusions

In this work, a meshless integral method based on the regularized boundary integral equation has been developed and applied to two-dimensional linear elasticity and elastoplasticity with small or large deformation. The meshless integral method is built on an earlier method proposed by Atluri and coworkers based on the local boundary integral equation representation for linear elasticity and incorporates improvements in four areas: First, a subtraction method is used successfully to remove the strong singularity in the governing equations and make the formula rather straightforward and more accurate in numerical solution than the direct limit approach used in [1], and simpler than the lengthy singularity removal treatment proposed in [2]. Secondly, accuracy is improved by employing a special numerical integration technique for the calculation of integrals with weak singularity (logarithmic type). A new method is also proposed to handle the essential boundary conditions explicitly, exactly, and efficiently. Finally, the natural boundary conditions are incorporated in the system governing equations and require no special handling.

The use of exact singular kernels (the fundamental solutions) contributes to the high accuracy of the governing boundary integral equations, while the singularity removal scheme introduced in this work makes it possible to solve the integral equation numerically with great precision. The method can be extended to 3D linear elasticity without major difficulties. Furthermore, this meshless method is truly meshless method and does not require a background mesh for interpolation or integration. Selected test problems were solved by this method and the results obtained were very accurate.

With the introduction of elastoplasticity with small deformation, an elastoplastic meshless integral method is obtained. The governing integral equation is obtained from the weak form of elastoplasticity with small deformation over a local sub-domain. The constitutive law used is rate-independent flow theory based on von Mises yielding criterion with isotropic hardening. Fixed point iteration (FPI) method is employed to solve the nonlinear governing equations. With the FPI method, the differentiation of the stiffness matrix is not needed, which

makes the formulation simple, and the implementation relatively easy. Finite element results are used as the bases for comparison with the meshless results. Numerical results show that this method can handle any prescribed loading profile, including unloading and reversed loading. This elastoplastic meshless method has shown good accuracy in numerical test problems (the constant patch tests, the shear patch test, the finite plate with a circular hole problem, and the thick-walled cylinder problem). In the thick-walled cylinder problem, it is shown that the sub-domain radius has a strong influence on the accuracy of numerical results, as does the yield criterion.

With the introduction of elastoplasticity with large deformation, a meshless method is obtained for analyzing large deformation of elastoplastic materials. Green-Naghdi elastoplastic theory is used for large deformation along with an updated Lagrangian description. The Green-Lagrange strain is related to the second Piola-Kirchhoff stress, and the yield function is constructed in the space of the second Piola-Kirchhoff stress. Fixed point iteration is employed to solve the governing equations, which are nonlinear because of the material nonlinearity (elastoplasticity) and geometric nonlinearity (large deformation). For problems with only large deformation, the results obtained were compared with the results from hand calculation solution and finite element analysis. For problems with both elastoplasticity and large deformation, the comparison is made between results from our meshless formulation and results from finite element analysis. The numerical results show excellent accuracy.

To use the meshless methods above, the user should generate an input file which contains the nodal information. A method based on generalized quadrilateral sectors is used to derive a pre-processor to assist the user in defining the data and discretization scheme for the proposed meshless method. The user can use the graphic user interface to enter the geometric model, as well as the definition of material, boundary conditions, and other parameters that are needed in the input file. A post-processor that is truly meshless is presented to enable users to analyze and visualize the results. In this post-processor, we traverse the device space pixel by pixel and for each pixel, we compute the point in the problem domain space that it maps to. Based on the computed value of the field variable of interest at this point, we set the color of the pixel. Using this method, we are able to completely eliminate the need for a mesh to support post processing. Results show that the pre-processor and the post-processor are effective and the latter is a computationally viable visualization tool for meshless methods.

Future research in this area could involve the extension of meshless integral method to perform analysis for 3D problems and coupling the meshless method with FEM. Most of the work related to the development of meshless method has been in 2D applications. It is clear that the potential of any meshless method lies in its possible application to 3D problems for which the discretization of the problem domain using traditional FEM is an expensive and tedious job. Meshless methods have shown great advantages over FEM, but the computational cost of a meshless method typically exceeds the cost of FEM. Furthermore, given the level of maturity and comprehensive capabilities of FEM, it is often advantageous to apply meshless method only in the local domains, where the advantages of meshless methods over FEM can be exploited to the fullest.

This research lays the foundation for modeling and simulation of metal cutting processes which are characterized by large deformation with elastoplasticity and moving boundary. This model could include the thermal effect of plastic deformation on the workpiece and tool. Using the augmented Lagrangian method, we could develop a contact algorithm, since the cutting process involves contact between the tool and the chip. The stick-slip formulation can be implemented for contact friction, using geometrical and/or physical criteria for detecting chip separation. The numerical solution of the resulting equations can be done using the central difference method, which is generally the method of choice for dynamic problems. Once models of the metal cutting process are developed using the meshless integral method, we will be able to predict the tolerance and residual stresses on the machining surfaces. This can be used as a basis for optimization of the cutting parameters, and for predicting the tool wear.

References

[1] Atluri S.N., Sladeck J., Sladeck V., and Zhu T. The local boundary integral equation (LBIE) and its meshless implementation for linear elasticity. Computational Mechanics. 2000; 25: 180-198.

[2] Sladek V., Sladek J., Atluri S.N., and Van Keer R. Numerical integration of singularities in meshless implementation of local boundary integral equations. Computational Mechanics. 2000; 25: 394-403.

Appendix A - Derivation of Meshless Integral Equation for Elastoplasticity with Small Deformation

In this section, we will explain the derivation of the equation 14 in Chapter 3.

We start with equation 4 in Chapter 3:

$$\int_{\Omega^{\langle a\rangle}} \left(\sigma_{ij,j}(\mathbf{x}) + b_i(\mathbf{x})\right) g_i\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) d\Omega(\mathbf{x}) = 0 \qquad \text{A.1}$$

In this work, following Atluri et al., we use a special test function defined by

$$g_i\left(\mathbf{x}, \mathbf{y}^{<a>}\right) = \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{<a>}\right) e_k\left(\mathbf{y}^{<a>}\right) = (u^*_{ki} - \tilde{u}_{ki}) e_k\left(\mathbf{y}^{<a>}\right) \qquad \text{A.2}$$

Here u^*_{ki} is the fundamental solution of elasticity and $\tilde{u}_{ki}$ is the companion solution. They are related to the stress fields σ^*_{ki} and $\tilde{\sigma}_{ki}$ that satisfy the following equations:

$$\sigma^*_{ik,k}\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) + \delta\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) e_i = 0 \qquad \text{A.3}$$

$$\begin{cases} \tilde{\sigma}_{ik,k} = 0 & \text{in } \left|\mathbf{x} - \mathbf{y}^{\langle a\rangle}\right| < h_s^{<a>} \\ \tilde{u}_{ki} = u^*_{ki} & \text{on } \left|\mathbf{x} - \mathbf{y}^{\langle a\rangle}\right| = h_s^{<a>} \end{cases} \qquad \text{A.4}$$

In A.3, $\delta(\mathbf{x}, \mathbf{y}^{<a>})$ is the Dirac delta function, and the differentiation operates on the field point $\mathbf{x}$. By construction, the special test function $\tilde{u}^*_{ji}$ is zero on the circle of radius $h_s^{<a>}$ centered at $\mathbf{y}^{<a>}$. With $\tilde{u}^*_{ji}$ as the test function, A.1 may be written as:

$$\int_{\Omega^{\langle a\rangle}} \left(\sigma_{ij,j}(\mathbf{x}) + b_i(\mathbf{x})\right) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) e_k d\Omega(\mathbf{x}) = 0 \qquad \text{A.5}$$

$$\int_{\Omega^{\langle a\rangle}} \sigma_{ij,j}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) e_k d\Omega(\mathbf{x}) + \int_{\Omega^{\langle a\rangle}} b_i(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) e_k d\Omega(\mathbf{x}) = 0 \qquad \text{A.6}$$

Because of the fact that:

$$\partial_j\left(\sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) e_k\right) = \sigma_{ij,j}(x) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) e_k + \sigma_{ij}(x) \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{\langle a\rangle}\right) e_k \qquad \text{A.7}$$

If we apply the second Green theorem to the left-hand side term of A.7:

$$\int_{\Omega} \partial_j \left(\sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k \right) d\Omega = \int_{\partial\Omega} \left(\sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k \right) n_j d\Gamma \tag{A.8}$$

A.7 and A.8 yield:

$$\begin{aligned} &\int_{\Omega} \left(\sigma_{ij,j}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{<a>}\right) e_k + \sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{<a>}\right) e_k \right) d\Omega \\ &= \int_{\partial\Omega} \left(\sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k \right) n_j d\Gamma \end{aligned} \tag{A.9}$$

$$\begin{aligned} &\int_{\Omega^{<a>}} \sigma_{ij,j}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k d\Omega \\ &= \int_{\partial\Omega} \left(\sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k \right) n_j d\Gamma - \int_{\Omega} \sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k d\Omega \end{aligned} \tag{A.10}$$

Similarly, we can write the same equation switching the test function and the non-test function and obtain the following equation:

$$\begin{aligned} &\int_{\Omega} \tilde{\sigma}^*_{ij,j}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) u_i(\mathbf{x}) d\Omega + \int_{\Omega} \tilde{\sigma}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) u_{i,j}(\mathbf{x}) d\Omega \\ &= \int_{\partial\Omega} \left(\tilde{\sigma}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) u_i(\mathbf{x}) \right) n_j d\Gamma \end{aligned} \tag{A.11}$$

Expressing σ_{ij} as:

$$\sigma_{ij} = \sigma^e_{ij} + \sigma^{cor}_{ij} = C^e_{ijkl} u_{k,l} + \sigma^{cor}_{ij} \tag{A.12}$$

$$\begin{aligned} &\int_{\Omega^{<a>}} \sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k d\Omega(\mathbf{x}) = \int_{\Omega^{<a>}} (C^e_{ijkl} u_{k,l} + \sigma^{cor}_{ij}) \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k d\Omega(\mathbf{x}) \\ &= \int_{\Omega^{<a>}} C^e_{ijkl} u_{k,l} \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k d\Omega(\mathbf{x}) + \int_{\Omega^{<a>}} \sigma^{cor}_{ij} \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k d\Omega(\mathbf{x}) \end{aligned} \tag{A.13}$$

and making use of the following equation (the Betty's reciprocal theorem):

$$\int_{\Omega^{\langle a \rangle}} \sigma_{ij}(\mathbf{x}) \tilde{u}^*_{ki,j}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) e_k d\Omega(\mathbf{x}) = \int_{\Omega^{\langle a \rangle}} \tilde{\sigma}^*_{ij}\left(\mathbf{x}, \mathbf{y}^{\langle a \rangle}\right) u_{i,j}(\mathbf{x}) d\Omega(\mathbf{x}) \tag{A.14}$$

We can rewrite A.13 as:

$$\int_{\Omega^{<a>}} \sigma_{ij}(\mathbf{x})\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)e_k\,d\Omega(\mathbf{x}) = \int_{\Omega^{<a>}} (C^{e}_{ijkl}u_{k,l} + \sigma^{cor}_{ij})\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)e_k\,d\Omega(\mathbf{x})$$
$$= \int_{\Omega^{<a>}} u_{k,l}\,\tilde{\sigma}^{*}_{kl}\,d\Omega(\mathbf{x}) + \int_{\Omega^{<a>}} \sigma^{cor}_{ij}\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)e_k\,d\Omega(\mathbf{x}) \qquad \text{A.15}$$
$$= \int_{\Omega^{<a>}} \tilde{\sigma}^{*}_{ij}\,u_{i,j}\,d\Omega(\mathbf{x}) + \int_{\Omega^{<a>}} \sigma^{cor}_{ij}\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)e_k\,d\Omega(\mathbf{x})$$

A.11 and A.15 yield:

$$\int_{\Omega^{<a>}} \sigma_{ij}(\mathbf{x})\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Omega(\mathbf{x}) = \int_{\partial\Omega}\left(\tilde{\sigma}^{*}_{ij}\left(\mathbf{x},\mathbf{y}^{<a>}\right)u_i(\mathbf{x})\right)n_j\,d\Gamma$$
$$- \int_{\Omega^{<a>}} \tilde{\sigma}^{*}_{ij,j}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)u_i(\mathbf{x})\,d\Omega + \int_{\Omega^{<a>}} \sigma^{cor}_{ij}\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Omega(\mathbf{x}) \qquad \text{A.16}$$

From A.12 and A.10, we can get:

$$\int_{\Omega^{<a>}} \sigma_{ij,j}(\mathbf{x})\,\tilde{u}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)e_k\,d\Omega = \int_{\partial\Omega^{<a>}}\left(\sigma_{ij}(\mathbf{x})\,\tilde{u}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)e_k\right)n_j\,d\Gamma$$
$$- \int_{\partial\Omega}\left(\tilde{\sigma}^{*}_{ij}\left(\mathbf{x},\mathbf{y}^{<a>}\right)u_i(\mathbf{x})\right)n_j\,d\Gamma + \int_{\Omega^{<a>}} \tilde{\sigma}^{*}_{ij,j}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)u_i(\mathbf{x})\,d\Omega \qquad \text{A.17}$$
$$- \int_{\Omega^{<a>}} \sigma^{cor}_{ij}\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Omega(\mathbf{x})$$

Taking into account the fact that $t_i = \sigma_{ij}n_j$, A.17 becomes:

$$\int_{\Omega^{<a>}} \sigma_{ij,j}(\mathbf{x})\,\tilde{u}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Omega = \int_{\partial\Omega^{<a>}} t_i(x)\,\tilde{u}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Gamma$$
$$- \int_{\partial\Omega^{<a>}} \tilde{t}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k u_i(\mathbf{x})\,d\Gamma + \int_{\Omega^{<a>}} \tilde{\sigma}^{*}_{ij,j}\left(\mathbf{x},\mathbf{y}^{<a>}\right)u_i(\mathbf{x})\,d\Omega \qquad \text{A.18}$$
$$- \int_{\Omega^{<a>}} \sigma^{cor}_{ij}\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Omega(\mathbf{x})$$

A.3, A.4, A.6 and A.18 together yield:

$$\int_{\partial\Omega^{<a>}} t_i(\mathbf{x})\,\tilde{u}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Gamma - \int_{\partial\Omega^{<a>}} \tilde{t}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k u_i(\mathbf{x})\,d\Gamma - \int_{\Omega^{<a>}} \delta\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_i u_i(\mathbf{x})\,d\Omega$$
$$- \int_{\Omega^{<a>}} \sigma^{cor}_{ij}\,\tilde{u}^{*}_{ki,j}\left(\mathbf{x},\mathbf{y}^{<a>}\right)e_k\,d\Omega(\mathbf{x}) + \int_{\Omega^{<a>}} b_i(\mathbf{x})\,\tilde{u}^{*}_{ki}\left(\mathbf{x},\mathbf{y}^{\langle a\rangle}\right)e_k\,d\Omega(\mathbf{x}) = 0 \qquad \text{A.19}$$

If we change the indices in A.19 so that all the components of $\mathbf{e}$ would be expressed with the same dummy index i, it leads to:

$$\int_{\partial\Omega^{<a>}} t_j(\mathbf{x})\tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})e_i d\Gamma - \int_{\partial\Omega^{<a>}} \tilde{t}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})e_i u_j(\mathbf{x})d\Gamma - \int_{\Omega^{<a>}} \delta(\mathbf{x},\mathbf{y}^{<a>})e_i u_i(\mathbf{x})d\Omega$$
$$- \int_{\Omega^{<a>}} \sigma^{cor}_{jk}\tilde{u}^*_{ij,k}(\mathbf{x},\mathbf{y}^{<a>})e_i d\Omega(\mathbf{x}) + \int_{\Omega^{<a>}} b_j(\mathbf{x})\tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})e_i d\Omega(\mathbf{x}) = 0 \qquad \text{A.20}$$

As e_i is arbitrary, we obtain the concluding equation:

$$\int_{\partial\Omega} t_j(\mathbf{x})\tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})d\Gamma - \int_{\partial\Omega} \tilde{t}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})u_j(\mathbf{x})d\Gamma - \int_{\Omega^{<a>}} \delta(\mathbf{x},\mathbf{y}^{<a>})u_i(\mathbf{x})d\Omega$$
$$- \int_{\Omega^{<a>}} \sigma^{cor}_{jk}\tilde{u}^*_{ij,k}(\mathbf{x},\mathbf{y}^{<a>})d\Omega(\mathbf{x}) + \int_{\Omega^{<a>}} b_j(\mathbf{x})\tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{\langle a\rangle})d\Omega(\mathbf{x}) = 0 \qquad \text{A.21}$$

After the removal of strong singularity, we obtain equation 3.14:

$$\alpha^{<a>}_{ij}u^{<a>}_j = -\int_{\Omega^{<a>}} \tilde{u}^*_{ij,k}(\mathbf{x},\mathbf{y}^{<a>})\sigma^{cor}_{ij}(\mathbf{x})d\Omega + \int_{\Gamma^{<a>}_u+\Gamma^{<a>}_t} \tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})t_j(\mathbf{x})d\Gamma$$
$$- \int_{C^{<a>}} \tilde{t}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})u_j(\mathbf{x})d\Gamma - \int_{\Gamma^{<a>}_u+\Gamma^{<a>}_t} \tilde{t}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})(u_j(\mathbf{x}) - u^{<a>}_j)d\Gamma \qquad \text{A.22}$$
$$+ \int_{\Omega^{<a>}} \tilde{u}^*_{ij}(\mathbf{x},\mathbf{y}^{<a>})b_j(\mathbf{x})d\Omega$$

Appendix B - Derivation of Meshless Integral Equation for Elastoplasticity with Large Deformation

In this section, we will explain the derivation of how to get the equation 14 in Chapter 4. We start with equation 4 in Chapter 4:

$$\int_{\Omega_0^{\langle a\rangle}} \left(\tau_{IJ,J}(\mathbf{X}) + \rho_0 f_I(\mathbf{X})\right) g_I\left(\mathbf{X}, \mathbf{Y}^{\langle a\rangle}\right) d\Omega_0(\mathbf{X}) = 0 \tag{B.1}$$

In this equation, $\Omega_0^{<a>} \subset \Omega_0$ is a sub-domain related to node <a>, $\mathbf{Y}^{<a>}$ is the position vector of node a which is also called a source point, $\mathbf{X}$ is the integration or field point which may or may not coincide with a node, and g_I is the test function. In this work, following Atluri et al. [16][17], we use a special test function defined by

$$g_I\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) = \tilde{u}^*_{JI}\left(\mathbf{X}, \mathbf{Y}^{<a>}\right) e_J\left(\mathbf{Y}^{<a>}\right) \tag{B.2}$$

where e_I represents the j-th component of a unit force vector, $\tilde{u}^*_{JI}$ is the special test function.

With the test function B.2, equation B.1 can be written as:

$$\begin{aligned}&\int_{\Omega_0^{<a>}} \tau_{IJ,J} \tilde{u}^*_{KI}(\mathbf{X}, \mathbf{Y}^{<a>}) e_K d\Omega_0 = \int_{\partial\Omega_0^{<a>}} \tau_{IJ}(\mathbf{X}) \tilde{u}^*_{KI}(\mathbf{X}, \mathbf{Y}^{<a>}) e_K n_J d\Gamma_0 \\ &- \int_{\Omega_0^{<a>}} \tau_{IJ}(\mathbf{X}) \tilde{u}^*_{KI,J}(\mathbf{X}, \mathbf{Y}^{<a>}) e_K d\Omega_0\end{aligned} \tag{B.3}$$

Similarly, we have:

$$\int_{\Omega_0^{<a>}} \tilde{\tau}^*_{IJ,J} u_I(\mathbf{X}) d\Omega_0 + \int_{\Omega_0^{<a>}} \tilde{\tau}^*_{IJ}(\mathbf{X}) u_{I,J}(\mathbf{X}) d\Omega_0 = \int_{\partial\Omega_0^{<a>}} \tilde{\tau}^*_{IJ}(\mathbf{X}, \mathbf{Y}^{<a>}) u_I \eta_J d\Gamma_0 \tag{B.4}$$

Where

$$
\begin{aligned}
&\int_{\Omega_0^{<a>}} \tau_{IJ,J}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&= \int_{\Omega_0^{<a>}} G^I_r(C^e_{LJMN}U_{M,N}\frac{\partial x_r}{\partial X_L})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&+ \int_{\Omega_0^{<a>}} G^I_r\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&= \int_{\Omega_0^{<a>}} \delta_{rI}(\delta_{rL}+\Delta u_{r,L})C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&+ \int_{\Omega_0^{<a>}} G^I_r\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&= \int_{\Omega_0^{<a>}} (\delta_{IL}+\delta_{rI}\Delta u_{r,L})C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&+ \int_{\Omega_0^{<a>}} G^I_r\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&= \int_{\Omega_0^{<a>}} C^e_{IJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&+ \int_{\Omega_0^{<a>}} \delta_{rI}\Delta u_{r,L}C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&+ \int_{\Omega_0^{<a>}} G^I_r\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&= \int_{\Omega_0^{<a>}} U_{M,N}\tilde{\sigma}^*_{MN}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) + \int_{\Omega_0^{<a>}} \delta_{rI}\Delta u_{r,L}C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \\
&+ \int_{\Omega_0^{<a>}} G^I_r\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})
\end{aligned}
\qquad \text{B.5}
$$

From B.5, we have:

$$\int_{\Omega_0^{<a>}} U_{M,N}\left[G_M^A\right]^{-1}\tilde{\tau}^*_{AN}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})+\int_{\Omega_0^{<a>}}\delta_{rI}\Delta u_{r,L}C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

$$+\int_{\Omega_0^{<a>}} G_r^I\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

$$=\int_{\Omega_0^{<a>}} U_{M,N}\tilde{\tau}^*_{MN}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})+\int_{\Omega_0^{<a>}}\delta_{rI}\Delta u_{r,L}C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

$$+\int_{\Omega_0^{<a>}} G_r^I\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) \qquad \text{B.6}$$

$$=\int_{\Omega_0^{<a>}} U_{I,J}\tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})+\int_{\Omega_0^{<a>}}\delta_{rI}\Delta u_{r,L}C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

$$+\int_{\Omega_0^{<a>}} G_r^I\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

For convenience, we denote:

$$\Lambda^{//}=\int_{\Omega_0^{<a>}}\delta_{rI}\Delta u_{r,L}C^e_{LJMN}U_{M,N}\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

$$+\int_{\Omega_0^{<a>}} G_r^I\frac{\partial x_r}{\partial X_L}T^{cor}_{LJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

So B.6 can be written as:

$$\int_{\Omega_0^{<a>}}\tau_{IJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})=\int_{\Omega_0^{<a>}} U_{I,J}\tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})+\Lambda^{//} \qquad \text{B.7}$$

B.4 and B.7 yield:

$$\int_{\Omega_0^{<a>}}\tau_{IJ}(\mathbf{X})\tilde{u}^*_{KI,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K\left(\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})=\int_{\partial\Omega_0^{<a>}}\tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})n_J d\Gamma$$

$$-\int_{\Omega_0^{<a>}}\tilde{\tau}^*_{IJ,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})+\Lambda^{//} \qquad \text{B.8}$$

Combining B.3 and B.8 we have following equation:

$$\int_{\Omega_0^{<a>}}\tau_{IJ,J}\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K d\Omega_0=\int_{\partial\Omega_0^{<a>}}\tau_{IJ}(\mathbf{X})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0$$

$$-\left[\int_{\partial\Omega_0^{<a>}}\tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})n_J d\Gamma-\int_{\Omega_0^{<a>}}\tilde{\tau}^*_{IJ,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})+\Lambda^{//}\right] \qquad \text{B.9}$$

B.1 and B.9 yield:

$$\int_{\partial\Omega_0^{<a>}} \tau_{IJ}(\mathbf{X})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 - \int_{\partial\Omega_0^{<a>}} \tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})n_J d\Gamma$$
$$+ \int_{\Omega_0^{<a>}} \tilde{\tau}^*_{IJ,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X}) - \Lambda^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_I(\mathbf{X})\,\tilde{u}^*_{KI}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K d\Omega_0(\mathbf{X}) = 0 \qquad \text{B.10}$$

B.10 can be rewritten as:

$$\int_{\partial\Omega_0^{<a>}} G^I_r(T_{LJ}(\mathbf{X})\frac{\partial x_r}{\partial X_L})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 - \int_{\partial\Omega_0^{<a>}} \tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})n_J d\Gamma$$
$$+ \int_{\Omega_0^{<a>}} \tilde{\tau}^*_{IJ,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X}) - \Lambda^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_I(\mathbf{X})\,\tilde{u}^*_{KI}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K d\Omega_0(\mathbf{X}) = 0 \qquad \text{B.11}$$

B.11 leads to:

$$\int_{\partial\Omega_0^{<a>}} G^I_r(T^0_{LJ}(\mathbf{X}) + \Delta T_{LJ}(\mathbf{X}))\frac{\partial x_r}{\partial X_L}\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 - \int_{\partial\Omega_0^{<a>}} \tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})n_J d\Gamma$$
$$+ \int_{\Omega_0^{<a>}} \tilde{\tau}^*_{IJ,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X}) - \Lambda^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_I(\mathbf{X})\,\tilde{u}^*_{KI}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K d\Omega_0(\mathbf{X}) = 0 \qquad \text{B.12}$$

Because $G^I_r = \delta_{rI}$, the first term of B.12 can be rewritten as:

$$\begin{aligned}
&\int_{\partial\Omega_0^{<a>}} G^I_r(T^0_{LJ}(\mathbf{X}) + \Delta T_{LJ}(\mathbf{X}))\frac{\partial x_r}{\partial X_L}\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 \\
&= \int_{\partial\Omega_0^{<a>}} \delta_{rI}(T^0_{LJ}(\mathbf{X}) + \Delta T_{LJ}(\mathbf{X}))(\delta_{rL} + \Delta u_{r,L})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 \\
&= \int_{\partial\Omega_0^{<a>}} \delta_{rI}T^0_{LJ}(\mathbf{X})(\delta_{rL} + \Delta u_{r,L})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 \\
&+ \int_{\partial\Omega_0^{<a>}} \delta_{rI}\Delta T_{LJ}(\mathbf{X})(\delta_{rL} + \Delta u_{r,L})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0
\end{aligned} \qquad \text{B.13}$$

Because $T^0_{LJ} = \sigma^0_{LJ}$, B.13 can be expressed as:

$$\begin{aligned}
&\int_{\partial\Omega_0^{<a>}} G^I_r(T^0_{LJ}(\mathbf{X}) + \Delta T_{LJ}(\mathbf{X}))\frac{\partial x_r}{\partial X_L}\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 \\
&= \int_{\partial\Omega_0^{<a>}} \sigma^0_{IJ}(\mathbf{X})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 + \int_{\partial\Omega_0^{<a>}} T^0_{LJ}(\mathbf{X})\Delta\Delta_{l,L}\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 \\
&+ \int_{\partial\Omega_0^{<a>}} \delta_{rI}\Delta T_{LJ}(\mathbf{X})(\delta_{rL} + \Delta u_{r,L})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0
\end{aligned} \qquad \text{B.14}$$

We denote:

$$\Lambda^{///} = \int_{\partial\Omega_0^{<a>}} T^0_{LJ}(\mathbf{X})\Delta u_{I,L}\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 + \int_{\partial\Omega_0^{<a>}} \delta_{rI}\Delta T_{LJ}(\mathbf{X})(\delta_{rL}+\Delta u_{r,L})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 \quad \text{B.15}$$

B.12, B.14, and B.15 give us:

$$\int_{\partial\Omega_0^{<a>}} \sigma^0_{IJ}(\mathbf{X})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 + \Lambda^{///} - \int_{\partial\Omega_0^{<a>}} \tilde{\tau}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})n_J d\Gamma + \int_{\Omega_0^{<a>}} \tilde{\tau}^*_{IJ,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X}) - \Lambda^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_I(\mathbf{X})\,\tilde{u}^*_{KI}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K d\Omega_0(\mathbf{X}) = 0 \quad \text{B.16}$$

Because $\tilde{\tau}^*_{IJ} = \tilde{\sigma}^{*0}_{IJ}$ we have:

$$\int_{\partial\Omega_0^{<a>}} \sigma^0_{IJ}(\mathbf{X})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K n_J d\Gamma_0 + \Lambda^{///} - \int_{\partial\Omega_0^{<a>}} \tilde{\sigma}^{*0}_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X})n_J d\Gamma + \int_{\Omega_0^{<a>}} \tilde{\sigma}^{*0}_{IJ,J}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X}) - \Lambda^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_I(\mathbf{X})\,\tilde{u}^*_{KI}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K d\Omega_0(\mathbf{X}) = 0 \quad \text{B.17}$$

With $T^0_I = \sigma^0_{IJ}n_J$, B.17 becomes:

$$\int_{\partial\Omega_0^{<a>}} T^0_I(\mathbf{X})\tilde{u}^*_{KI}(\mathbf{X},\mathbf{Y}^{<a>})e_K d\Gamma_0 + \Lambda^{///} - \int_{\partial\Omega_0^{<a>}} \tilde{t}^*_{KI}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K u_I(\mathbf{X})d\Gamma + \int_{\Omega_0^{<a>}} \delta\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_I u_I(\mathbf{X}) - \Lambda^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_I(\mathbf{X})\,\tilde{u}^*_{KI}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_K d\Omega_0(\mathbf{X}) = 0 \quad \text{B.18}$$

If we change the indice in B.18 so that all the components of **e** would be expressed with the same dummy index i, B.18 leads to:

$$\int_{\partial\Omega_0^{<a>}} T^0_J(\mathbf{X})\tilde{u}^*_{IJ}(\mathbf{X},\mathbf{Y}^{<a>})e_I d\Gamma_0 + A^{///} - \int_{\partial\Omega_0^{<a>}} \tilde{t}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_I u_J(\mathbf{X})d\Gamma + \int_{\Omega_0^{<a>}} \delta\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_I u_I(\mathbf{X}) - A^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_J(\mathbf{X})\,\tilde{u}^*_{IJ}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_I d\Omega_0(\mathbf{X}) = 0 \quad \text{B.19}$$

Where $A^{//}$ and $A^{///}$ are derived from $\Lambda^{//}$ and $\Lambda^{///}$ respectively:

$$A^{//} = \int_{\Omega_0^{<a>}} \delta_{rJ}\Delta u_{r,L}C^e_{LKMN}U_{M,N}\tilde{u}^*_{IJ,K}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_I d\Omega_0(\mathbf{X}) + \int_{\Omega_0^{<a>}} \delta_{rJ}\frac{\partial x_r}{\partial X_L}T^{cor}_{LK}(\mathbf{X})\tilde{u}^*_{IJ,K}\left(\mathbf{X},\mathbf{Y}^{<a>}\right)e_I d\Omega_0(\mathbf{X})$$

$$A^{///} = \int_{\partial\Omega_0^{<a>}} T_{LK}^0(\mathbf{X})\Delta u_{J,L}\tilde{u}_{IJ}^*(\mathbf{X},\mathbf{Y}^{<a>})e_I n_J d\Gamma_0$$
$$+ \int_{\partial\Omega_0^{<a>}} \delta_{rJ}\Delta T_{LK}(\mathbf{X})(\delta_{rL}+\Delta u_{r,L})\tilde{u}_{IJ}^*(\mathbf{X},\mathbf{Y}^{<a>})e_I n_K d\Gamma_0$$

As e_i is arbitary, we obtain the following equation:

$$\int_{\partial\Omega_0^{<a>}} T_J^0(\mathbf{X})\tilde{u}_{IJ}^*(\mathbf{X},\mathbf{Y}^{<a>})d\Gamma_0 + B^{///} - \int_{\partial\Omega_0^{<a>}} \tilde{t}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_J(\mathbf{X})d\Gamma$$
$$+ \int_{\Omega_0^{<a>}} \delta\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_I(\mathbf{X}) - B^{//} + \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_J(\mathbf{X})\,\tilde{u}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) = 0 \qquad \text{B.20}$$

Where $B^{//}$ and $B^{///}$ are derived from $A^{//}$ and $A^{///}$ respectively:

$$B^{//} = \int_{\Omega_0^{<a>}} \delta_{rJ}\Delta u_{r,L} C_{LKMN}^e U_{M,N}\tilde{u}_{IJ,K}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$
$$+ \int_{\Omega_0^{<a>}} \delta_{rJ}\frac{\partial x_r}{\partial X_L}T_{LK}^{cor}(\mathbf{X})\tilde{u}_{IJ,K}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X})$$

$$B^{///} = \int_{\partial\Omega_0^{<a>}} T_{LK}^0(\mathbf{X})\Delta u_{J,L}\tilde{u}_{IJ}^*(\mathbf{X},\mathbf{Y}^{<a>})n_J d\Gamma_0$$
$$+ \int_{\partial\Omega_0^{<a>}} \delta_{rJ}\Delta T_{LK}(\mathbf{X})(\delta_{rL}+\Delta u_{r,L})\tilde{u}_{IJ}^*(\mathbf{X},\mathbf{Y}^{<a>})n_K d\Gamma_0$$

After removing strong singularity using subtraction method, we obtain equation 4.14:

$$\alpha_{IJ}^{<a>}u_J^{<a>} = \int_{\Gamma_u^{<a>}+\Gamma_t^{<a>}} T_J^0(\mathbf{X})\tilde{u}_{IJ}^*(\mathbf{X},\mathbf{Y}^{<a>})d\Gamma_0 + B^{///} - \int_{C^{<a>}} \tilde{t}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)u_J(\mathbf{X})d\Gamma$$
$$- \int_{\Gamma_u^{<a>}+\Gamma_t^{<a>}} \tilde{t}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)(u_J(\mathbf{X}) - u_J^{<a>})d\Gamma_0 - B^{//} \qquad \text{B.21}$$
$$+ \int_{\Omega_0^{\langle a\rangle}} \rho_0 f_J(\mathbf{X})\,\tilde{u}_{IJ}^*\left(\mathbf{X},\mathbf{Y}^{<a>}\right)d\Omega_0(\mathbf{X}) = 0$$

Appendix C - Publications and Presentations

1. Jianfeng Ma**,** X. J. Xin, and Prakash Krishnaswami. Meshless Integral Method for Elastoplastic Materials with Large Deformation**.** International Journal for Numerical Methods in Engineering. (to be submitted)
2. Jianfeng Ma**,** X. J. Xin, and Prakash Krishnaswami. Elastoplastic Meshless Integral Method. Computer Methods in Applied Mechanics and Engineering. (to be submitted)
3. Jianfeng Ma**,** X. J. Xin, and Prakash Krishnaswami. A Meshless Integral Method and Its Application to Elastoplastic Problems. 7th World Congress on Computational Mechanics. Los Angeles, California. July 16-22, 2006.
4. Jianfeng Ma**,** X. J. Xin, and Prakash Krishnaswami. *Meshless Integral Method with Mixed Boundary Conditions in Elasticity*. 11th Annual Graduate Research Forum. Kansas State University, March 3, 2006.
5. Jianfeng Ma, Prakash Krishnaswami, and X. J. Xin. *A Truly Meshless Pre- and Post-Processor for Meshless Analysis Methods*. Advances in Engineering Software. 2007(38), 9-30.
6. Jianfeng Ma, Prakash Krishnaswami, and X. J. Xin. *A New and Efficient Pixel-Based Post-Processor for Meshless Method.* 8th US National Congress on Computational Mechanics. Austin, Texas. July 25-27, 2005.
7. Jianfeng Ma**,** X. J. Xin, Anthony Bodin, and Prakash Krishnaswami. *Nodal Density Refinement and Adaptivity for the Meshless Integral Method*. 27th World Conference on Boundary Elements and other Meshless Reduction Methods. Orlando, Florida. March 15-17, 2005.
8. Anthony Bodin, Jianfeng Ma, X. J. Xin, and Prakash Krishnaswami. *A Meshless Integral Method Based on Regularized Boundary Integral Equation*. Computer Methods in Applied Mechanics and Engineering, v195, n44-47, September 15 2006, p6258-6286.
9. Jianfeng Ma, X. J. Xin, Anthony Bodin, and Prakash Krishnaswami. *The Meshless Integral Method: The Effects of Support Domain, Sub-domain, Nodal Density and Monomial Basis*. 7th US National Congress on Computational Mechanics. Albuquerque, New Mexico, July 27-31, 2003.

Appendix D - Honors and Awards

1. K-State MNE Outstanding Graduate Researcher Award. Department of Mechanical & Nuclear Engineering, Kansas State University. April 24, 2007.
2. Graduate Research Assistantship/Teaching Assistantship. Department of Mechanical & Nuclear Engineering, Kansas State University. 2002-2007.
3. Financial assistance of 7th World Congress on Computational Mechanics. Los Angeles, California. July 16-22, 2006
4. Financial assistance of 8th US National Congress on Computational Mechanics. Austin, Texas. July 24-28, 2005.
5. Graduate Travel Award of National Science Foundation Experimental Program to Simulate Competitive Research, Kansas. 2004
6. Fellowship Award of 7th US National Congress on Computational Mechanics. Albuquerque, New Mexico. July 27-31, 2003.